THE DESIGN OF COLLABORATIVE PROJECTS:

Language, Metaphor, Conversation and the Systems Approach

By

JULIAN DAY

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Management Engineering
(Faculty of Engineering & The Built Environment)

UNIVERSITY OF CAPE TOWN

April 1999

Supervisor: Dr Johan Strümpfer

22

THE DESIGN OF COLLABORATIVE PROJECTS:

Language, Metaphor, Conversation and the Systems Approach

By

JULIAN DAY

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Management Engineering
(Faculty of Engineering & The Built Environment)

UNIVERSITY OF CAPE TOWN

April 1999

Supervisor: Dr Johan Strümpfer

THE DESIGN OF COLLABORATIVE PROJECTS:

Language, Metaphor, Conversation and the Systems Approach

By: JULIAN DAY

ABSTRACT

This thesis uses a systems approach to develop a model for Collaborative Project Design (CPD). Failure of the software process is the area of concern. The focus of the argument is, however, on the organizational environment of the software process. A central argument is that the analytic tools of standard software development methodologies are inappropriate for systems synthesis. They provide little assistance in coping with the loose complexity that is inherent in the organizational environment in which the software process is embedded. These analytic tools and the engineering language and metaphor which dominate the software process undermine collaboration and disempower business users. CPD was developed to enable viable collaboration that is necessary for the software process to succeed. The purpose of CPD is to provide a systemic model of causal influences and social process in order to guide a project designer when intervening in projects which call for acts of shared creation and/or discovery. CPD was developed through a combination of action research (in projects involving software development and organisational transformation) and theoretical readings focused on the philosophy of meaning, systems thinking, social process and the software process. CPD emphasises that a collaborative project requires careful design of its underlying languages, metaphors and conversations. It identifies three distinct types of conversation, namely communication, dialogue and collaboration. The thesis describes how these conversation types are utilised in transforming a project's network of commitments from loose complexity via shared meaning to cohesive simplicity. Associated with each conversation type is a set of project influences which are developed into a causal influence model in order to depict a collaborative project as a dynamic system of mutually interdependent influences. This causal influence model was used to synthesise the learning from action research and the theoretical readings. An appreciative systems framework was then derived in order to justify a collaborative project as a self-regulating social system and was overlaid onto the causal influence model in order to derive CPD in its final form. CPD proved beneficial when tested in practical projects as a framework to organise a project designer's mind when designing project interventions.

Address: P.O.Box 48230
Kommetjie, 7975
Cape Town
South Africa

Telephone: (021) 7833519 (h)
(021) 7833520 (w)
Fax: (021) 7833520
Email: dayjules@iafrica.com

CONTENTS

PREFACE: A PERSONAL HISTORY INTO THE THESIS		**4**
1.	**INTRODUCTION AND OVERVIEW**	**10**
1.1	CATEGORIES FOR THE DESIGN OF COLLABORATIVE PROJECTS	14
1.2	CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS	19
1.3	THE APPRECIATIVE SYSTEMS FRAMEWORK	22
1.4	MODEL FOR THE DESIGN OF COLLABORATIVE PROJECTS (CPD)	24
1.5	OVERVIEW SUMMARY	26
2.	**THE RESEARCH AREA OF CONCERN**	**27**
2.1	INTRODUCTION	27
2.2	THE SOFTWARE CRISIS	28
2.3	COMPARISON BETWEEN HUMAN POPULATED SYSTEMS AND COMPUTER SYSTEMS	29
2.4	THE SOFTWARE PROCESS	31
2.5	I.T.'S RESPONSE TO THE SOFTWARE CRISIS	32
2.6	THE ENVIRONMENTAL FALLACY	34
2.7	METAPHOR AND THE SOFTWARE PROCESS	36
2.8	SOFTWARE DEVELOPMENT AS ORGANISATIONAL CONVERSATION	37
2.9	TRANSFORMING THE I.T. LANGUAGE GAME	38
2.10	THE RESEARCH QUESTION	39
2.11	THE RESEARCH METHOD	40
3.	**CATEGORIES FOR THE DESIGN OF COLLABORATIVE PROJECTS**	**45**
3.1	INTRODUCTION	45
3.2	ACTING	45
3.3	THINKING	56
3.4	CONVERSING	72
3.5	THE ACT MOVEMENT PROJECT	81
4.	**CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS**	**91**
4.1	INTRODUCTION	91
4.2	THE SOCIOLOGY OF KNOWLEDGE AND THE SOCIAL CONSTRUCTION OF REALITY	93
4.3	SOCIAL PROCESS AND THE DIALECTIC OF KNOWLEDGE	95
4.4	THE VOCABULARY OF THE SOCIOLOGY OF KNOWLEDGE	96
4.5	THE CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS	103
4.6	THE COMMON STRUCTURE OF THE SOCIAL PROCESS SHARED BY SUBSYSTEMS	109
5.	**THE APPRECIATIVE SYSTEMS FRAMEWORK**	**112**
5.1	INTRODUCTION	112
5.2	OVERVIEW OF APPRECIATIVE SYSTEMS	112
5.3	NOTATATION OF THE APPRECIATIVE SYSTEMS FRAMEWORK	114
5.4	THE APPRECIATIVE SYSTEMS FRAMEWORK	117
5.5	THE APPRECIATIVE SYSTEM AND COLLABORATIVE PROJECTS	120
5.6	JUSTIFYING 5 KEYWORDS AS INTRA-SUBSYSTEM MOVEMENT NAMES	121
6.	**A MODEL FOR THE DESIGN OF COLLABORATIVE PROJECTS**	**124**
6.1	DERIVATION OF THE CPD MODEL	124
6.2	JUSTIFYING 9 KEYWORDS AS INTER-SUBSYSTEM MOVEMENT NAMES	124
6.3	THE STRUCTURE OF CPD	127
6.4	USING CPD IN PRACTICE	127
7.	**ACTION RESEARCH**	**131**
7.1	PROJECT 1 - THE ACT MOVEMENT PROJECT	131
7.2	PROJECT 2 - THE CORE/SATELLITE 'CRISIS'	133
7.3	PROJECT 3 - RELATIONAL DATABASE	141
7.4	PROJECT 4 - ADMIN TRANSFORMATION	151
7.5	PROJECT 5 - NEW DEALING SYSTEM	156
7.6	PROJECT 6 - THE PROPHET PROJECT	164
7.7	CONCLUSION - ANSWERING THE RESEARCH QUESTION	171
8.	**CONCLUSION AND CRITIQUE**	**175**
9.	**REFERENCES**	**181**
10.	**BIBLIOGRAPHY**	**186**
APPENDICES AND FOLD-OUTS OF MODELS		**203**

PREFACE: A PERSONAL HISTORY INTO THIS THESIS

This research thesis has come about through a long struggle. I have worked for seventeen years in software development for various commercial organisations as a programmer, systems analyst, business analyst, project manager, mentor, trainer, facilitator, manager and researcher.

During this period, developing software in commercial organisations has become increasingly problematic. My experience drives me to the conclusion that programming, while far from trivial, is not overwhelmingly difficult. Learning a new technology is, also, not essentially a problem. Providing that time and care is devoted to the learning process, most programmers are keen and able to learn the new technologies that the technical specialists are so prolific in inventing.

The methods and principles associated with systems analysis cause more difficulties. Data modelling, for example, is a standard method in most systems development methodologies. Structured Systems Analysis and Design Method (SSADM), for example, calls its data models "Logical Data Structures" (Longworth and Nicholls, 1986, page LDS1 to LDS21), while Information Engineering (IE) calls them "Entity Relationship Diagrams" (Martin, 1987, page 213). Whatever the name, there are essentially a few standard methods for modelling abstract data structures. Similarly, there are standard methods for process modelling and program specification. As analysis techniques, there is no doubt that they are useful. If a system exists in some form so that it is possible to be analysed, and if one was interested in analysing its data structure, then standard data modelling methods would prove to be extremely useful. But in my capacity as a "systems analyst", my experience leads me to conclude that I am seldom called on to *analyse* a system. I am more likely to be called on when a new system is desired, or an old one is collapsing, and especially when computers are involved in some way. "Systems analyst" as a job title or role is a misnomer because there is seldom a pre-existing system 'there' to be analysed. A more appropriate title is "systems synthesist" (Weinberg, 1988, page 9).

An example of systems analysis would be automating a manual payroll system. Here a system exists. It is 'there' to be analysed. Its data structures, for example, can be analysed and modelled. The business system is not essentially being changed. Tax will still be calculated, pay cheques will still be issued, and records of hours worked and leave due will still be kept. Some human tasks will fall away and be replaced by computer processes. The major adaptation for the humans will be to observe the automated system and make sure that it works at least as well as the manual system it replaced. As a "systems analyst" I have a full skills set of effective analytical methods, tools and techniques to accomplish such a task.

But tasks are not usually presented in this way. Far more likely is that there is a problem, an idea, a crisis, a hope, a threat, an opportunity or some other pressing need accompanied by a feeling that computer technology is either responsible or involved in some way. These will require solutions that

go beyond mere analysis. Something new will have to be created, built, designed, or as Weinberg suggests, "synthesised" (Weinberg, 1988, page 11). A new system will invariably be created, or an old one re-designed. This will affect both human systems and computer systems which are to be interwoven into a whole, integrated system. The essential point is that the business *will* fundamentally change. Tasks which require systems analysis are trivial compared with tasks that require systems synthesis.

An example might be receiving news that a competitor has just implemented a new system whereby it can sell its product over the Internet allowing it to reduce its prices, offer better service, cut out intermediaries and participate in a wider market. Another example might be that product development has just launched a new product which the current infrastructure can not handle. Another example is that a software vendor has announced that it will no longer support the old software packages which support the organisation's mission critical operations. The list of examples is endless.

The information systems (I.S.) methodologies that helped me to be competent as an analyst provide very little assistance in helping me to become competent as a synthesist. Most of these methodologies assume "business requirements" (Checkland and Holwell, 1998, page 122) and then provide tools for modelling them. But users normally do not have a well-defined set of requirements, only problems and hopes. A chasm now separates the so-called 'business user' from the so-called 'I.T. professional'. A task that requires mainly synthesis is delegated to a profession skilled mainly in analysis. In theory, it would be possible at this point to turn the business user away by pointing out that it is inappropriate to expect a profession with analysis skills to perform a function of synthesis. We could point out, for example, that the technological experts who design and manufacture motor vehicles are not routinely called upon to solve single-handedly the problem of urban traffic congestion through their engineering expertise alone. If they are called upon to assist in such a matter, it is to add their technological expertise as participants in a wider conversation. Such a conversation involves synthesising a solution and it would be unusual to entrust them with the ultimate responsibility for both designing the conversation and solving the problem. But the I.T. industry *does not* turn the business user away. It *does* attempt to perform the function of synthesis using only analytical tools, and it *does* often end up carrying responsibility for the whole process. From this confusion, a dysfunctional software process has become habitualised as a standard form of life.

As a practitioner in the I.T. industry, I am in no position to turn my business users away. It would be deemed unprofessional. Initially it seemed that the problem must lie in my mastery of the analysis skills set, but it soon became apparent that it is the skills set itself that is lacking. In particular, tasks requiring systems synthesis typically require collaborative conversation, but the I.T. skills set is impoverished in this regard. For example, JAD (joint application development) workshops initially raised some hope in their attempt to encourage proactive user involvement in the software process. They represented an overt acknowledgement that systems synthesis is a real problem. But, having

facilitated many JAD sessions, I conclude that they add no fundamental new element to the software process. JAD merely provides a forum, but analysis tools are still used for a process that requires synthesis. JAD was an overt attempt by I.T. to design a conversation that would enable synthesis, but it was founded on the same confusion that pervades the software process in general. The arena in which the systems analyst operates is this uncomfortable chasm of confusion. The analyst is expected to bring sanity to this part of the software process, but without any effective skills to do so. The analyst's portion of the software process is not viable and this disrupts the whole development process. Project management by implication becomes equally non-viable because the process being managed is flawed.

In order to cope, I concluded that it was necessary to develop an entirely new type of skill, which is what this research thesis attempts to do. It represents the attempt of an I.T. practitioner to redevelop coping skills in an industry that is confused. Checkland and Holwell in the first part of their book "Information, Systems, and Information Systems: making sense of the field" elaborate on the confused state of the industry. They point out the inadequacy of I.T. textbooks and state that the conventional story presented to college students is a long way from the day-to-day reality which an I.T. professional experiences (Checkland and Holwell, 1998, page 121). It is too late at this stage to turn the business user away, as we might have done in the past. We have allowed expectations about what we can deliver to remain artificially high, even in the face of so much failure. We have effectively disempowered the business users who have never learned to compensate for our limitations. Instead, we need some new skills in 'turning around' the situations with which we are presented so that synthesis is possible.

These difficulties prompted me to look outside I.T. for help. Systems thinking seemed to have much to offer concerning the synthesis and design of complex systems. As Checkland and Holwell point out in discussing I.S. (information systems):

"...it seemed remarkable that an area called 'information *systems*' made very little use of systems ideas or the process of thinking systemically" (ibid., page xiv)

My experience as a JAD facilitator focused my attention on conversation design. I.T.'s best attempt at conversation design was flawed. The ability to run a JAD session successfully is well outside the capability of the average systems analyst. The chances of success are slim. A group of diverse people attempt to *synthesise* on the fly in a restricted time period a complex system comprising people and computers using *analysis* techniques; the combined output is modelled in an abstract modelling language that business users do not fully comprehend. The analyst not only understands all the diverse business vocabularies, appreciates all the various business problems, but can, on the fly, translate and re-represent them as a synthesised whole in an abstract modelling language. On top of this the analyst can facilitate the workshop, handle group conflict, and know what questions to ask

next. JAD attempted to promote collaboration, but often promoted irritation. I wanted to find ways of designing conversations where true collaboration, essential to the software process but usually absent, could be routinely achieved by average practitioners.

This interest in conversation design focused attention on language. I became convinced that partnership between business user and software developer would never be achieved as long as I.T. language dominated the conversations. As Checkland and Holwell point out there is a "system which serves" and "a system which is served" (ibid., page 111). Why is the dominant language not the language of the business user or the system to be served? I eventually concluded that, deep in the conceptual system of both I.T. professional *and* business user, was the assumption that, fundamentally, software development was an engineering process and that therefore I.T. language was appropriate. This focused attention on metaphor. It became apparent that the industry was dominated by an engineering metaphor. The engineering metaphor was so strong that it was reflected in our language, our methods and coloured the whole software process. But there are other metaphors, and there is a danger in being consistently trapped by only one. I concluded that an organisation and therefore the software process could be redescribed metaphorically as a conversation.

As a practitioner I began to operate differently, consciously thinking about the design of the conversations going on around me. I tried to find ways of getting users to collaborate with me, to help me build things in a common language rather than just answer questions which I then modelled. I became more conscious of listening to their language to see if I could represent things in ways we could both understand. Metaphorically, I saw the organisation as a complex conversation as opposed to a production line. I started to see that my analysis tools were private methods that could seldom be used as shared language between business user and software developer. I started to gain an ability to 'turn situations around' so that collaboration could happen. Some of the successes seemed significant enough to require further research.

This thesis represents research into the cluster of related conclusions discussed above. This research is essentially that of a practitioner who has turned to theory outside of his normal environment in order to pursue what seem like promising practical possibilities. My essential thesis is that systems thinking combined with insights into language, metaphor and conversation can be synthesised into a systemic model that will facilitate the design of collaborative projects. If we are able to design collaborative projects successfully, then we will, amongst other things, contribute greatly to the success of the software process. The initial area of concern is the software process and its history of failure, but the arguments focus on collaboration. The reader may feel that the thesis is disjointed in this respect. It may seem that one question is posed, but a different one is answered. I need to emphasise that I remain a practitioner as opposed to a philosopher, but in order to make sense of my practical problems, I am forced to look outside normal boundaries and consider the environment

which contains the problem. Checkland and Holwell have developed the POM model (processes for organisational meaning) which emphasises the organisational setting, social process, discourse, creating meaning, purposeful action, and appreciative systems (Checkland and Holwell, 1998, page 106). This effectively embeds I.T. and I.S. in their wider context. My aims are complementary to this, but my model-building focuses on the design of collaboration. The software process is the area of concern and also the area of application, but the main contribution and arguments relate to the environment of the software process. Synthesis of any kind of system, not just computer systems, involves conversation. The design of these conversations is the focus of this thesis.

Collaboration is not, therefore, just an I.T. problem. If there was a sound tradition whereby commercial organisations knew when and how to design for collaboration, I believe that the software process would not be as impoverished as it is at the moment. Business users would have noticed that we have a limited skills set, would have understood that this skills set does not lend itself to collaboration, and would have used *their* collaborative expertise to 'turn *us* around', to turn systems analysis into systems synthesis. But they have not done this because they also do not have this expertise. My conclusion is that the first step in bridging the chasm between systems developer and business user is to develop a general organisational ability to design collaborative conversations. My essential argument is that collaboration is essential for synthesising human systems, computer systems and, when necessary, integrating both types of system into a cohesive whole. As a practitioner before I ask "What are your requirements?" I need to have answered "How is it possible for us to talk?". My hope is that in the future, business users will ask *me* that latter question, and if I have no satisfactory response, will have in *their* skills sets an ability to design a conversation suitable for our collaboration.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Johan Strümpfer, for his sincere encouragement and support, for his wisdom, and for conceiving a PhD programme which stretched my learning experience far beyond my expectations.

I would like to thank Tom Ryan of the University of Cape Town for his dedicated support of students involved in the Programme for Systems Management.

I would like to thank Warren Topp, a fellow PhD student, for his energetic and considered input into our group discussions, for pointing me to relevant literature for my research, and for his general encouragement and support.

CHAPTER 1. INTRODUCTION AND OVERVIEW

This thesis has been written as a result of the worldwide software crisis (DeMarco, 1995, page 139) characterised by ever increasing failure of the software process whereby business application systems are developed and implemented in organisational settings. Information technology (I.T.) and information systems (I.S.), although separate fields, are not often sharply differentiated and the terms are used interchangeably (Checkland and Holwell, 1998, page 8 to 11). The software process involves both fields because typically it is concerned with implementing information systems via information technology, specifically computer technology. The acronym 'I.T.' will be used as an umbrella term to refer to the whole industry including information systems. 'Software process' will be used more specifically to refer to the process whereby an organisation implements computer software, which it may itself have developed, to serve its business systems.

As will be argued in detail in Chapter 2, my essential argument is that the software crisis is caused largely by communication problems directly related to I.T.'s limited set of skills. Specifically, analytic tools are habitually used for functions that require synthesis. This undermines an ability to collaborate and therefore inhibits the whole software process. The problem area of concern, therefore, is the software process. My argument is that a model that enables the design of successful collaborative projects will contribute greatly to resolving the software crisis (See Chapter 2 for full argument).

This chapter gives an overview of the model building process involved in this thesis. No substantiated or referenced arguments are developed at this stage and the reader should not be perturbed by this because all terms and model building elements introduced in Chapter 1 will be described, substantiated and referenced in subsequent chapters. The reader should merely get a pre-orientation of the direction in which the model building will proceed and the elements involved. Thus, the first purpose of this chapter is to provide a framework to help orientate the reader in order to facilitate the reading and assimilation of the whole thesis. Schema 1 supports this purpose by providing a schematic to outline the logical structure of the thesis. A reader can refer back to this schematic at any time during the reading of the thesis in order to understand where a particular argument, model or set of vocabulary fits into the logical development of the whole thesis. The second purpose of this chapter is to present a condensed overview or summarised pre-reading of the argument, model and vocabulary of the thesis so that when detailed, substantiated and fully referenced arguments are developed in later chapters, there is a pre-orientation as to where these arguments fit into the whole. Forward referencing from the condensed overview (this chapter) to the detailed arguments of later chapters, sections or sub-sections in the main body of the thesis should help to orientate the reader. These two purposes should mutually support each other; an understanding of the structure of argumentation should help the assimilation of content and a pre-orientation of the content should help clarify the logic of argumentation, model building and development of the vocabulary of collaborative projects.

ARGUMENT

Collaborative project
"ACTS of SHARED CREATION"
Derive 3 domains & 3 subsystems
Describe Table 1 structure, category types
Chapter 1

Step 1
Categories for the design of collaborative projects
Develop theoretical and philosophical positions for each domain and category type.
Derive a set of distinctions.
Develop a set of categories and insights to aid the designer.
Justify 12 influencers.
Chapter 3

Step 2
Causal influences of collaborative projects
Derive the social process underpinning collaborative projects and represent it as a system of causal influences using influencers identified in Table 1.
Identify the common structure of the 3 subsystems.
Derive measurabilty for 12 influencers.
Introduce 14 Keywords.
Chapter 4

Step 3
Appreciative systems framework
Justify collaborative projects as appreciative systems. Link the appreciative systems framework to the social process and causal influences developed via Fig 1.
Overlay the common structure of the 3 subsystems w ith the appreciative systems framework.
Justify 5 intra-subsystem movements.
Justify 4 influencer types.
Chapter 5

Step 4
Model for collaborative project design
Extend the appreciative systems framework by overlaying it onto the full set of causal influences represented in Fig 1.
Justify 9 inter-subsystem movements.
Chapter 6

MODEL BUILDING

3 DOMAINS OF COLLABORATIVE PROJECTS

ACTING (Acts)	THINKING (Creation)	CONVERSING (Shared)

18 category types

TABLE 1

CATEGORY TYPE	ACTING	THINKING	CONVERSING
Domain			
Conversation type			
Process			
Intervention			
Influencer	red	green	blue

FIG 1

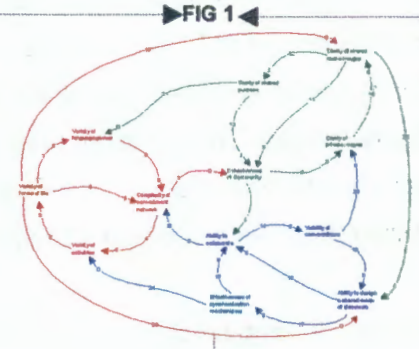
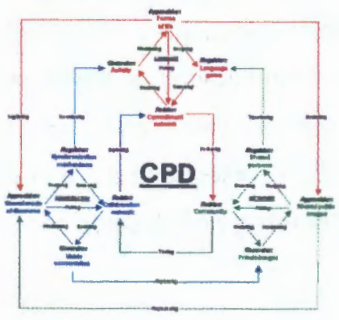


FIG 2

FIG 3



VOCABULARY

3 SUBSYSTEMS:
Language (Human activity)
Metaphor (Conceptual)
Conversation (Interpretative)

12 INFLUENCERS:
Activities
Forms of life
Commitment networks
Language games
Private images
Shared public images
Community
Shared purpose
Viable conversation
Shared mode of discourse
Collaboration network
Synchronization mechanisms

14 KEYWORDS:
Socializing
Worldmaking
Forming
Designing
Governing
Legitimizing
Normalizing
Performing
Trusting
Representing
Transforming
Expressing
Explicating
Co-ordinating

12 MEASURABLE INFLUENCERS:
Variety of activities
Variety of forms of life
Complexity of commitment network
Variety of language games
Clarity of private images
Clarity of shared public images
Cohesiveness of community
Clarity of shared purpose
Viability of conversations
Ability to design a shared mode of discourse
Ability to collaborate
Effectiveness of synchronization mechanisms

4 INFLUENCER TYPES:
Generator
Appreciator
Relater
Regulator

5 INTRA-SUBSYSTEM MOVEMENTS:
Socializing
Worldmaking
Forming
Designing
Governing

9 INTER-SUBSYSTEM MOVEMENTS:
Legitimizing
Normalizing
Performing
Trusting
Representing
Transforming
Expressing
Explicating
Co-ordinating

Schema 1 - Outline of thesis structure

Schema 1 depicts four steps culminating in the derivation of a model for collaborative project design (CPD). Each step corresponds to a chapter (Chapters 3, 4, 5 and 6) which develops arguments, builds a vocabulary and consolidates the arguments and the vocabulary into a model. Each step builds a model that is utilised by the following step. The models are built up from arguments and a vocabulary derived from those arguments. Chapter 7 details action research conducted to test the models.

Fold-outs for Table 1, Fig 1, Fig 2 and Fig 3 are included in the appendices at the very end of the thesis. A fold-out for Scehma 1 is provided inside the front cover. This will enable the reader to have the critical models available without having to continually page through the text to refer to the models.

Step 1 - Categories for the design of collaborative projects

A table of categories is built, Table 1, which differentiates three domains within a collaborative project (refer to fold-out of Schema 1 and fold-out of Table 1). The *structure* of this table is described below (see section 1.1 for overview of table *contents*). Each domain has four associated category types:

- the **domain emphasis** highlights the critical focus of the domain
- the domain is associated with and manifest as a particular **subsystem**
- the **subsystem name** exemplifies a critical theoretical underpinning of the subsystem which is also related to the research topic (see thesis title)
- **grounding** identifies the main theoretical or practical position from which the arguments within each domain are developed

Each domain is associated with a particular conversation type which has four associated category types:

- the **conversation focus** is aligned to and supports the corresponding domain emphasis
- the **conversation type** is the most suitable form of conversation to support the conversation focus
- the conversation type utilises, attends to and is characterised by specific **conversation elements**
- the **conversation metaphor** provides a means of exemplifying the nature of the conversation type and allowing it to be conceptualised metaphorically

Each domain relates to a stage in the process of designing collaborative projects. The process category types represent four sets of positions in the form of distinctions relative to each domain on **mental representations**, **location of intelligence**, **background** and **process**. The conversation type categories and the process categories mutually support each other. The latter provide critical insights to organise the designer's mind when considering how to design the movement from 'loose complexity' through 'shared meaning' to 'cohesive simplicity'.

Each domain advocates **designer's interventions** and **designer's challenges**. This set of category types builds on all the insights and distinctions within a domain to assist with the design of appropriate interventions in collaborative projects.

Finally, each domain develops a set of influencers which are utilised in subsequent model building. The category types of **generator, appreciator, relater** and **regulator** are derived in Chapter 5 from the appreciative systems framework. The influencers are differentiated by (arbitrary) colours of red, green and blue in order to assist tracking the influencers throughout the various stages of model building. (See section 1.1 for an overview and Chapter 3 for full development of the content of Table 1.)

Step 2 - Causal influences of collaborative projects.

A causal influence model is developed, Fig 1, which builds on Step 1 by relating the influencers of Table 1 into a dynamic whole system. The interrelationships between the influencers are developed by understanding the social process which constructs the reality of a collaborative project. The vocabulary is extended by associating each influence between and within subsystems with a keyword that represents the essence of the influence. This entails deriving measures for the 12 influencers. (See section 1.2 for an overview and Chapter 4 for full development of the argument.)

Step 3 - The appreciative systems framework.

The previous step highlighted a common structure and common social process shared by the three subsystems. Step 3 derives an appreciative systems framework and overlays this on the common structure of the subsystems. It extends the vocabulary by deriving from the appreciative systems framework 4 influencer types and also by justifying 5 keywords as intra-subsystem movements. (See section 1.3 for an overview and Chapter 5 for full development of the argument.)

Step 4 - Model for collaborative project design (CPD).

A final model is derived, Fig 3, by using the causal influence model of Step 2 to expand the appreciative systems framework of Step 3. CPD emerges as a systemic lens through which to view the design of a collaborative project. It integrates into a system the categories that make up the vocabulary of a collaborative project. It highlights simultaneously both a system of influences and a social process through which designers can look to organise their minds when designing or intervening in collaborative projects. Step 4 extends the vocabulary by justifying 9 keywords introduced in Step 2 as inter-subsystem movements. (See section 1.4 for an overview and Chapter 6 for full development of the argument.)

1.1 CATEGORIES FOR THE DESIGN OF COLLABORATIVE PROJECTS

This section sets out categories and distinctions within collaborative projects and can be viewed in Table 1. This overview will emphasise the connection between the text and Table 1 by highlighting in **bold type** Table 1 categories which appear in the text. It will also provide forward references to later sections and subsections where the categories are developed in full. (Refer to fold-out of Schema 1 and fold-out of Table 1.)

This thesis is concerned with collaborative projects or "acts of shared creation and/or shared discovery" (Schrage, 1995, page 4). This phrase implies three domains: "creation and discovery" imply **thinking** (section 3.3), "shared" implies **conversing** (section 3.4), and "acts" implies **acting** (section 3.2). Individuals through their imaginings generate new thoughts or images. These may be expressed and shared with others. These shared thoughts and images may then inspire a group of individuals to perform co-ordinated activity in order to achieve their common purpose. Each domain thus emphasises a specific aspect of collaborative projects, namely **imagining**, **expressing** and **performing**. 'Imagining' focuses attention on shared **conceptual systems** (subsection 3.3.4) and the manner in which individual **private images** (subsections 3.3.1 and 4.3.4) generate **shared public images** (subsection 4.3.4) capable of uniting a **community** (subsection 3.3.1) around a **shared purpose** (subsection 3.3.1). 'Expressing' focuses attention on shared **interpretative systems** (Vickers, 1968, page xiii) which are needed to make conversation viable (**viable conversation**) (subsection 3.4.9) and to provide a **shared mode of discourse** (subsection 3.4.9) through which a **collaboration network** (subsection 3.4.5) can be synchronized (**synchronization mechanisms**) (subsection 3.4.6). 'Performing' focuses attention on **human activity systems** (Checkland, 1981, page 110) and the manner in which individual **activities** (subsection 3.2.1) and language (**language games**) (subsection 3.2.1) intertwine to produce shared **forms of life** (subsection 3.2.1) manifested as networks of mutual commitment (**commitment networks**) (subsection 3.2.6). These three systems constitute three interrelated subsystems of collaborative projects. They require a project designer to focus on three critical issues. The conceptual system requires a project design that generates shared **knowledge** (subsection 3.3.1and 4.3.4) through a group **learning** (subsection 3.3.3) process. The interpretative system requires a project design that enables individuals, often from widely divergent backgrounds, to be able to design for themselves a system that provides a **basis for viable, understandable and meaningful conversations** (subsection 3.4.9). The human activity system requires a project design that enables **expectations** (subsection 3.2.7) of **behaviour** (subsection 3.2.7) and performance to be clearly explicated.

This conception of collaborative projects highlights three domains that provide a basis for a more extensive categorization scheme. It is important to emphasise that my concern is with the *design* of collaborative projects and its three associated types of conversation. A focus on *conversation* design

uncovers some challenges and paradoxes which traditional project management overlooks and which this thesis will attempt to uncover.

CATEGORIES	ACTING	THINKING	CONVERSING
Domain emphasis	Performing	Imagining	Expressing
Subsystem	Human activity system	Conceptual system	Interpretative system
Subsystem name	Language	Metaphor	Conversation
Grounding	Philosophy of meaning	Systems thinking	Action research
Conversation focus	Behaviour Expectations	Knowledge Learning	Basis for viable, understandable, and meaningful conversations
Conversation type	Communication	Dialogue	Collaboration
Conversation elements	Speech acts Final Vocabulary	Images and Metaphors Assumptions	Modes of representation Re-representations
Conversation metaphor	"World re-maker"	"Socio-therapy"	"Conversation choreography"
Process	Loose complexity	Shared meaning	Cohesive simplicity
Mental representations	No representations	Multiple representations	Off-loaded representations
Location of intelligence	Individual Expert	Group Collective intelligence	Environment Designed, reusable mind tools
Background	Taken for granted Ready-to-hand	Unfolded and reflected upon Present-to-hand	Explicated as foreground Manipulable representations
Designer's interventions	Observation and description Coordination and languaging	Systems approach Metaphoric redescription	Shared space design Conversation design
Designer's challenges	Conduit metaphor "Submarine analogy"	Humiliation and cruelty Ironic attitude	Representation paradox Private methods
Generator	Activities	Private images	Viable conversations
Appreciator	Forms of life	Shared public images	Shared modes of discourse
Relater	Commitment networks	Community	Collaboration Networks
Regulator	Language games	Shared purpose	Synchronization mechanisms

Table 1. Categories for the design of collaborative projects

Part of the purpose of this thesis is to emphasise the usefulness to the practitioner of designing with a conscious awareness of the roles of **language** (subsection 3.2.1), **metaphor** (subsection 3.3.4) and **conversation** (subsection 3.4.6) in organisational life. **Metaphoric redescription** (subsection 3.3.5) becomes a powerful means of reframing conceptual systems and is useful to the systems designer as a method of inquiry. The designer needs insights into the nature of language because of its power to govern and transform human activity systems. By observing language, a window is provided into the conceptual system which is structured by "deep metaphors" (subsection 3.3.5) that govern behaviour. When meaningful conversation breaks down, as so often happens, the designer needs to enable the

design of a conversation that allows project members to create a mode of discourse that will re-establish viable conversation.

An organisation is viewed as an ongoing network of conversation (subsection 3.2.6) comprising many types of conversation. A collaborative project, as a type of organisation, needs to be designed so that the three critical domains of thinking, conversing and acting can interact harmoniously. The domain of thinking requires the design of a conversation that focuses on the conceptual system and the generation of shared knowledge. This is achieved through **dialogue** (subsection 3.3.3) which has **images** (subsection 3.3.1), **metaphors** (subsection 3.3.4) and **assumptions** (subsection 3.3.3) as its main elements required to build shared meaning. The domain of conversing requires the design of a conversation that focuses on the interpretative system and the generation of a shared basis for viable, understandable and meaningful conversation. This is achieved through **collaboration** (subsection 3.4.10) which has explicit representations (**modes of representation)** (subsection 3.4.8) and **re-representations** (subsection 3.4.2) as its main elements required to build a shared mode of discourse. The domain of acting requires the design of a conversation that focuses on the human activity system and the generation of clear expectations of behaviour and performance. This is achieved through **communication** (subsection 3.2.7) which has **speech acts** (subsection 3.2.6) and a shared vocabulary (**final vocabulary**) (subsection 3.3.6) as its main elements required to build mutual commitments.

Because organisations are not typically viewed as conversations, there has generally been little attempt to make sharp distinctions between conversation types. Eskimos have many different words for 'snow' because the distinctions between different types of snow are important to their projects. My contention is that similar distinctions between conversation types are important to *our* projects. At the moment, the word most commonly used is 'communication', a catchall term which covers a wide range uses. Failure of communication is usually cited as a cause of project failure, especially failed software projects (section 2.2). A close look at the linguistic and metaphoric underpinnings of the word 'communication' show that it has a rather limited use. This thesis explores three conversation types in relation to the design of collaborative projects and it limits them to very specific uses. No attempt is made to identify *all* types of organisational conversation.

Collaborative projects are fundamentally concerned with meaning. The software process is conceived as a journey from **'loose complexity'** through **'shared meaning'** to **'cohesive simplicity'** (Chapter 2 and section 6.4). The domain of acting is grounded in a position based on the **philosophy of meaning**. **Systems thinking** is fundamentally concerned with building shared meaning and provides the grounding for arguments in the domain of thinking. The domain of conversation is grounded primarily in insights gained from **action research** conducted in alignment with the metaphoric conception of an organisation as a conversation. (Action research is documented in section 3.5 and Chapter 7 and the action research approach is outlined in section 2.11)

The philosophy of meaning holds that we do not function in the world primarily by having mental representations of it (**no representations**) (subsection 3.2.2). Most of our activity takes place against a background where objects and properties are unconsciously **taken for granted** (subsection 3.2.2). An expert engaged in familiar activity does not have a representation of that activity 'in the mind'. Everything is simply 'there' and working, it is **ready-to-hand** (subsection 3.2.2), and is not consciously thought about. In the event of some breakdown, one begins to think about the activity and to make mental representations of it. The taken-for-granted background is **unfolded and reflected upon** (subsection 3.3.2) and made consciously present 'in the mind' (**present-to-hand**) (subsection 3.3.2).

Breakdowns can often be represented from multiple perspectives (**multiple representations**) (subsection 3.3.2). Complex representations may be off-loaded from the mind in some tangible form such as a map, diagram, tool, or model. Explicit representations allow the sharing of complex thoughts and images. They are also conducive to manipulation (**manipulable representations**) (subsection 3.4.7). A group that uses an **off-loaded representation** (subsection 3.4.1) as a collaboration tool is effectively working in a designed foreground (**explicated as foreground**) (subsection 3.4.7). Gradually, if expertise develops after habitual use, this foreground may become taken for granted and fade into background. In the domain of acting, intelligence is located in the **individual** (subsection 3.2.2) who is an unconsciously skilled **expert** (subsection 3.2.2). Observers notice the manner in which experts' activity is co-ordinated through communication. When there is a breakdown of meaningful communication, we attempt to tap into a **collective group intelligence** (subsection 3.3.3) through dialogue. The shared meaning which this produces may be too complex to hold 'in the mind', so the collective intelligence is off-loaded as a designed representation onto the **environment** (subsection 3.4.1) where it can be used as a **designed, resuable mind-tool** (subsection 3.4.1) in a collaboration.

A designer who is concerned with collaboration will observe human activity systems in order to determine the effectiveness of their functioning (**observation**) (subsection 3.2.3). The designer will observe the way that activity is co-ordinated through speech acts and the nature of the mutual commitments which they realise (**coordination**) (subsection 3.2.7). In particular, the designer will observe **languaging** (subsection 3.2.7), that is communication *about* communication, the way in which actors maintain or enhance their ability to communicate and thus to co-ordinate their human activity systems. In making observations, the designer can describe these human activity systems (**description**) (subsection 3.2.3), and thus make distinctions that experts within the system may not make. For example, a submarine navigator who has never been outside his submarine but is skilled at avoiding obstacles may take evasive action on account of some blips on a sonar screen. An observer, watching from a distance, notices a submarine avoiding a pod of whales. But the navigator cannot make sense of the observer's description, firstly because this activity is not differentiated from

any other normal activity in navigating the submarine, and secondly because there is no differentiated concept or word such as whale, rock, reef, or even submarine in this closed but functioning world. Activity is co-ordinated through communication and thus governed by language. The common conception of communication is founded on a widespread but unsatisfactory metaphor, the **conduit metaphor** (subsection 3.2.8), which sees communication as a transmission of information. This assumes that sender and receiver have a shared system for interpreting their communications and that communication is therefore fairly easy because it is merely a matter of packaging a thought into words and sending it. An alternative metaphor is proposed which acknowledges that communication requires much more effort than the conduit metaphor suggests. Systems of interpretation often break down. The **submarine analogy** (subsection 3.2.5) provides an example. The **'world re-maker'** (subsection 3.2.9) metaphor acknowledges that human communication requires a continual making and re-making of a shared interpretative system. In order for the submarine navigator and the submarine observer to communicate, they will have to work hard at reconstructing each other's worlds so that there is a shared context for mutual interpretation.

The designer of collaborative projects will have to be able to cope with situations characterized by critical breakdown of the interpretative system. In these situations there is a critical loss of meaning and an inability to interpret effectively the speech acts which co-ordinate mutual commitments. Dialogue is an appropriate form of conversation for unfolding and re-establishing shared meaning. Metaphorically, dialogue is like **'socio-therapy'** (subsection 3.3.3). Assumptions underlying private and communal thought are literally 'suspended' for observation allowing a **group** (subsection 3.3.3) to tap into a **collective intelligence** (subsection 3.3.3). The **systems approach** (subsection 3.3.1) seeks to build meaning through comprehensive representation of the problem situation. There is an awareness of the design of the method of inquiry into a problem situation and the influence that the design has on the type and organisation of knowledge or meaning which it produces. The systems approach seeks to sweep in as many variables and perspectives as it can in order to determine the shared issues and purpose around which a cohesive community can form. Problem situations are often underpinned by a deep metaphor that can be discovered by observing the language that is spoken. For example, the dominant language of I.T. is engineering (section 2.7). This thesis came about through metaphoric redescription of *the software process as a conversation that needs to be designed*. **Metaphoric redescription** (subsection 3.3.5) can provide new meanings to old issues and adds a creative element to inquiry, but it also introduces a challenge. Certain people have an **ironic attitude** (subsection 3.3.6) because they realise that anything can be made to look good or bad simply by being redescribed, but to most people, being redescribed by someone else is potentially humiliating and cruel (**humiliation and cruelty**) (subsection 3.3.7), and thus undermines the trust required for collaboration.

In order to move toward a conversation type of collaboration, the designer will endeavour to create a "shared space" (**shared space design**) (subsection 3.4.3). This will include a meaningful mode of

representing a thought, object, scheme, plan, event and so on. A foreground object will be created that is manipulable and has built-in intelligence. A group can work in shared space in order to create something in common. The designer will avoid the trap of using an incomprehensible **private method** (subsection 3.4.4) which is understood by only a sub-section of the group. This cannot act as shared space because it will undermine collaboration and introduce a language that cannot be interpreted by the whole group. A complex collaboration may require a number of shared spaces to be synchronised by conscious design (**conversation design**) (subsection 3.4.10). There will certainly need to be a learning mechanism somewhere in the collaboration system whereby the activity in one sub-collaboration enhances the ability of another sub-collaboration. The collaboration must be 'choreographed' so that sub-collaborations support each other (**'conversation choreography'**) (subsection 3.4.6). Certain collaborations are easy to design in the sense that there are ready-made modes of representation (e.g. buildings plans, spreadsheets) that can be utilised as shared space. But often the mode of representation will have to be designed from scratch which is difficult. There is now a **representation paradox** (subsection 3.2.4 and 3.4.8) to be confronted: experts do not have mental representations and may have no ready-made distinctions for designing a mode of representation; beginners or people confused by a problem situation will be likely to have inaccurate mental representations or, as observers of experts, may make distinctions that experts find incomprehensible.

1.2 CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS

Fig 1 uses the three sets of influencers from Table 1 to depict the causal influences of collaborative projects. (Refer to fold-out of Schema 1 and fold-out of Fig 1.) The influencers are extended by adding measurablility (e.g. influencer 'activities' is extended to become 'variety of activities' which implies that the amount of variety can be measured in some way). Each influence is numbered for convenient referencing and will be briefly commented upon to clarify aspects of the influence. Keywords, added in brackets, highlight the essence of each influence and contribute to an understanding of the social process underlying collaborative projects. (Sections 4.2, 4.3 and 4.4 describe the social process in detail. Section 4.5 derives from the social process a system of interrelated influences and introduces the keywords. These keyword names are finally justified in sections 5.6 and 6.2).

3 central inter-subsystem influences:

1. A cohesive community builds mutual trust which enhances its ability to collaborate. **(Trusting)**
2. A group that can collaborate effectively is able to explicate meaningful commitments by virtue of a shared interpretative system. **(Explicating)**
3. A commitment network which regularly meets the expectations that it sets will strengthen the cohesiveness of the community it serves. The more complex or fragmented the commitment network,

the more difficult it is to set realistic expectations and to perform according to those expectations. **(Performing)**

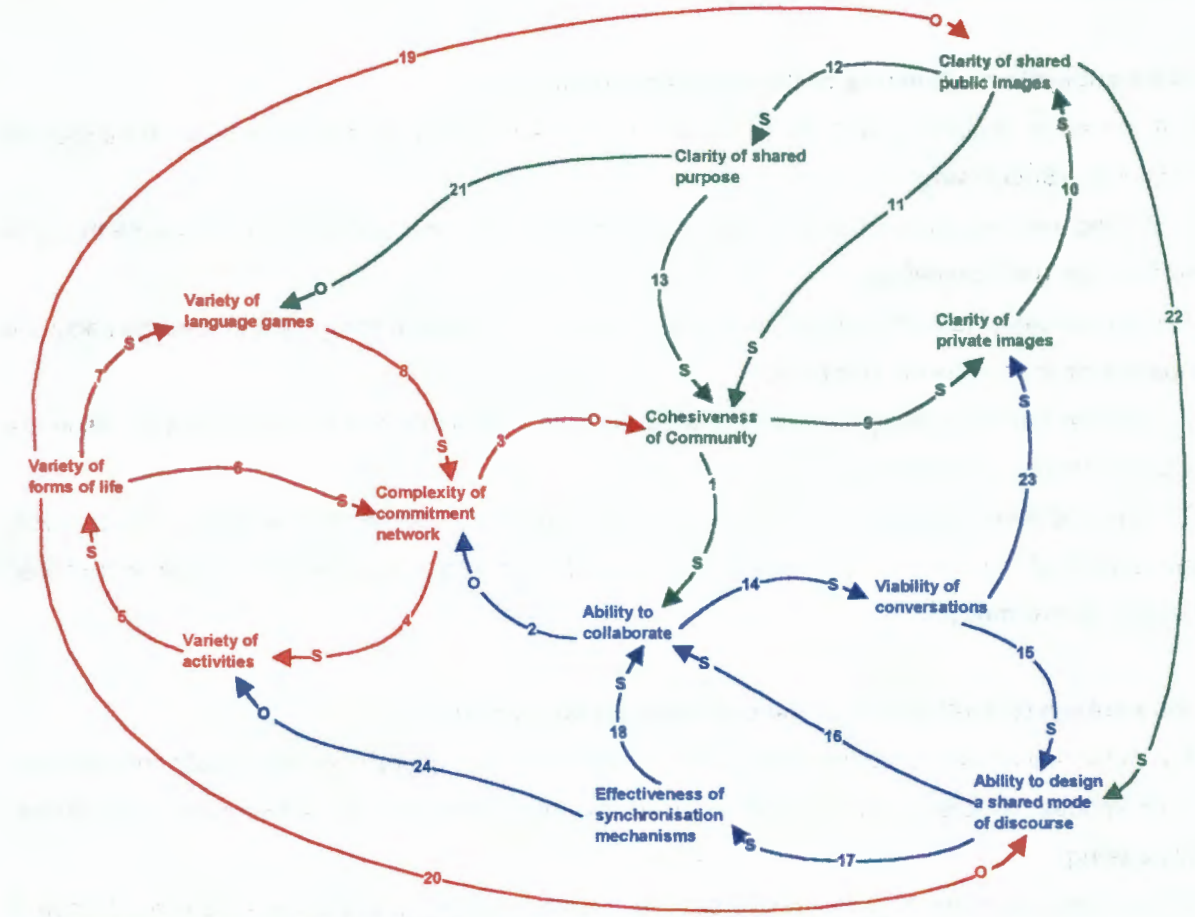


Fig 1. Causal influences of collaborative projects

5 intra-subsystem influences of the language subsystem:

4. The number and variety of individual activities are determined by the number and complexity of commitments entered into. Regularised commitment networks socialise new members to perform regularised activities. **(Socialising)**

5. Certain habitual activities may become combined and regularised into discernible forms of life. The more activities there are, the more numerous and varied are these forms of life. **(Worldmaking)**

6. Each form of life is co-ordinated via speech acts which form networks of commitment. As the variety of forms of life increase, so the network of mutual commitments becomes more complex. **(Forming)**

7. Each form of life creates or designs its own vocabulary and private methods, in other words, its own language games. As the variety of forms of life increases, so too do the associated language games. **(Designing)**

8. Language games govern which speech acts are possible and therefore what commitments can be made. As the variety of language games increases, an increasingly complex network of commitments becomes possible. **(Governing)**

5 intra-subsystem influences of the metaphor subsystem:

9. A cohesive community will be more effective in socialising its members than a fragmented community. **(Socialising)**

10. Shared meaning will be easier to unfold if individuals are clear about their own private thoughts and feelings. **(Worldmaking)**

11. It will be easier to form a cohesive community around an issue if thoughts and feelings about the issue are clear and shared. **(Forming)**

12. It will be easier to design a clearly shared purpose if thoughts and feelings about an issue are clear and shared. **(Designing)**

13. The cohesiveness of the community will be strengthened if its members share a clear purpose. Governance of the community is easier if it is conducted in accordance with a clear and shared purpose. **(Governing)**

5 intra-subsystem influences of the conversation subsystem:

14. A network that can collaborate effectively will increase the viability of its individual conversations. Viable modes of discourse can be re-used by new members or subsequent generations. **(Socialising)**

15. The more viable the individual conversations, the more ability there is to design a shared mode of discourse. **(Worldmaking)**

16. A shared mode of discourse enables individuals to become related by forming a network of collaborators who share a meaningful interpretative system. **(Forming)**

17. A shared mode of discourse allows 'conversation choreography'. Synchronisation mechanisms can be designed to co-ordinate the network of collaborators. **(Designing)**

18. Effective synchronisation mechanisms integrated into a well-designed coordination system will facilitate governance of the collaboration network. **(Governing)**

6 inter-subsystem movements:

19. Forms of life normalise thoughts and feelings about certain issues. As the variety of forms of life increases, the less likely it is that there will be a clear and shared view on any particular issue. **(Normalizing)**

20. Forms of life sanction their own specific modes of discourse. For example, the construction industry legitimizes architectural drawings as shared space for collaborations concerning the design and construction of houses. The more numerous and varied the forms of life, the more difficult it will be to find a standard, shared mode of discourse. **(Legitimizing)**

21. A clear and shared purpose provides an opportunity to limit the variety of language games to only those which clearly support the purpose. A leader or designer can transform the language game which governs the commitment network by re-aligning the language game to suit and support the purpose. **(Transforming)**

22. A clearly shared public image which can be suitably represented and off-loaded will increase the ability to design a shared mode of discourse. **(Representing)**

23. If conversations are not viable, it will be difficult for an individual to express her private images or make them clear to others. **(Expressing)**

24. Synchronisation mechanisms are designed to limit wasteful or counter-productive activities by conscious design of a system of learning and coordination mechanisms. **(Co-ordinating)**

1.3 THE APPRECIATIVE SYSTEMS FRAMEWORK

In this step, I derive an appreciative systems framework by synthesising the causal influences of collaborative projects as represented in Fig 1 and Vickers' conception of "appreciative systems" (Section 5.2, 5.3 and 5.4). (Refer to fold-out of Schema 1 and fold-out of Fig 2.) Influences 4 through 18 share a common structure and common keywords (Section 4.6). By re-representing the influences depicted in Fig 1, this commonality can be seen in Fig 2. At the same time the system of influences can be enriched by overlaying an appreciative systems framework which categorises influences and influencer types. Fig 2 reminds us that while it may be useful at times to see the distinctions between the domains of acting, thinking and conversing, they are in reality an inseparable unity. Thus, acting, thinking and conversing are not separable in reality but maybe separable in our mental models during the process of reflecting on reality. The appreciative system is a social system comprising two sub-systems. There is a **main system** which may form fairly naturally, and a **control system** which may be designed to deliberately govern the main system. The system is self-regulating in that it sets its own standards from within as a result of its ongoing social functioning.

The main system has an *influencer type* of **relater** which is a network of various networks of relationships which link individuals together. There is a **socializing** *movement* which culturally shapes the individual. There is an *influencer type* of **generator** which emphasises that individuals makes active choices about what they think or say or do. There is a **worldmaking** *movement* in which individuals combine to create shared meaning and to make common sense of the world. There is an *influencer type* of **appreciator** whereby individuals share common meaning and collectively come to see the world in certain specific ways to the exclusion of many others. There is a **forming** *movement* in which various networks of relationships come about as a result of the co-operative action fostered by a shared appreciation of the world.

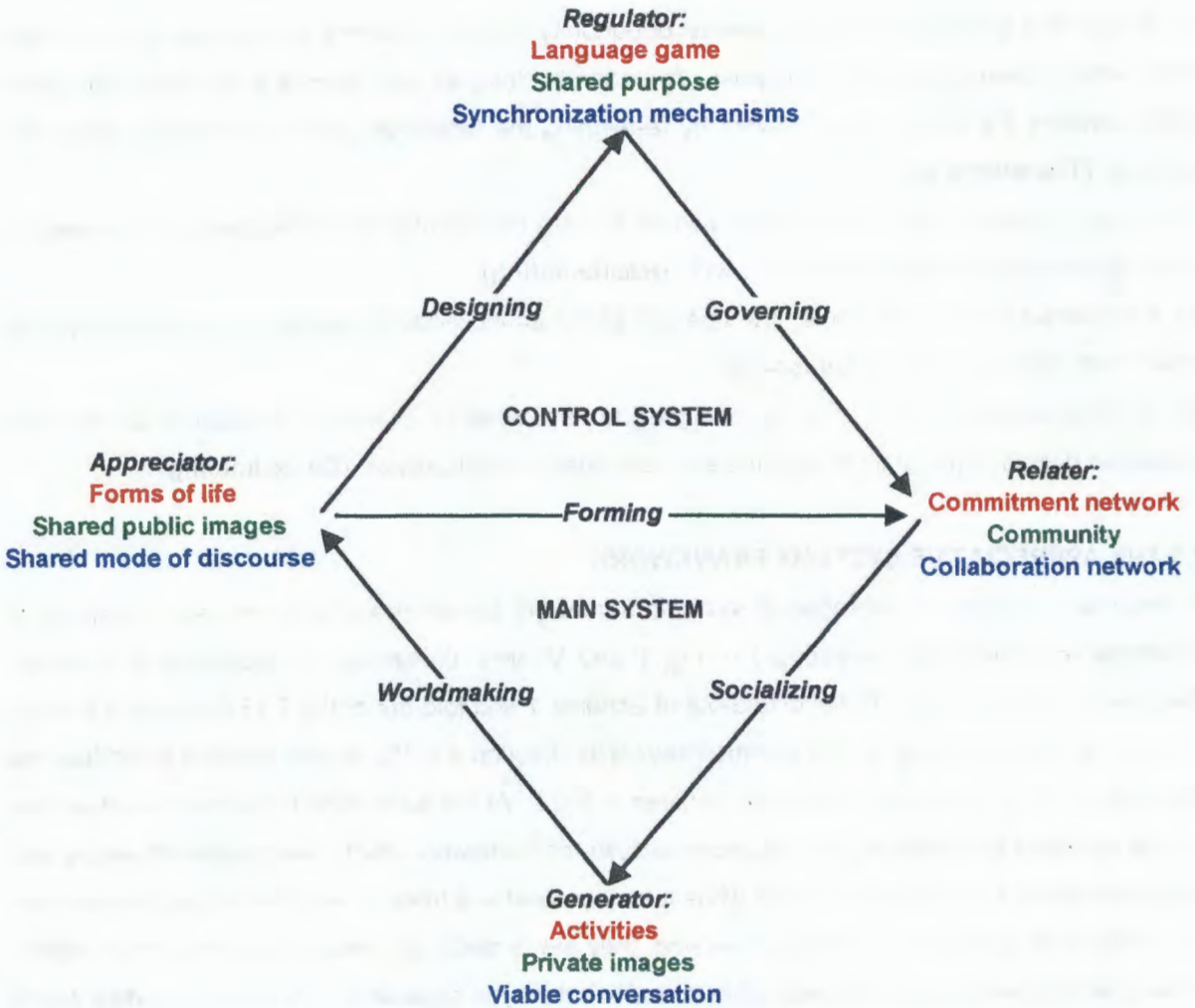


FIG 2. Appreciative systems framework

A control system may be designed to regulate the main system. There is a **designing** *movement* in which both the mechanisms for making collective choices and the collective choices themselves combine in a process of policy formulation. There is an *influencer type* of **regulator** which comprises standards and system settings of sufficient coherence to enable deliberate regulation of the system. There is a **governing** *movement* which detects disparity between relater and regulator, and triggers the activity necessary to realign the relater to the settings of the regulator.

Individuals through their **activities**, imaginings **(private images)** and conversation **(viable conversation)** may come to see the world in similar ways. An appreciation of their common **forms of life**, shared ideas **(shared public images)** and mutual way of talking **(shared mode of discourse)** allows them to form natural relationships with each other. They collaborate **(collaboration network)** in making commitments **(commitment network)** to each other. Successfully performing these commitments strengthens a sense of **community** and trust. These relationships may need deliberate regulation requiring the design of special vocabularies and methods **(language games)**, an explicit **shared purpose** and deliberate coordination and **synchronisation mechanisms**. The community

then socialises new generations into its accepted ways of acting, thinking and conversing. The system becomes unstable if individuals generate too many new choices causing the system of interpretation to break down.

1.4 MODEL FOR THE DESIGN OF COLLABORATIVE PROJECTS (CPD)

The appreciative systems framework depicted influencers and movements *within* each of three subsystems. Fig 1 shows that there are however another nine influences which act *between* the three subsystems. This section incorporates these remaining influences by expanding the appreciative systems framework shown in Fig 2. The full model for collaborative project design is represented in Fig 3. (Refer to fold-out of Schema 1 and fold-out of Fig 3.)

CPD is a lens through which to look at a collaborative project. It contains three interrelated subsystems: Language, Metaphor and Conversation. Each subsystem in turn has a main system and a control system. CPD poses 36 questions that enables a designer to organise her mind when designing or intervening in a collaborative project. It poses a number of questions about the state of the twelve influencers. For example, how many forms of life are involved in a project, how related are these forms of life, what language game dominates the project, how cohesive is the commitment network, what modes of discourse are available, how shared or clear is purpose and so on. It poses questions about how these influencers came to be in various states. Thus, each of the 24 movements represents a design issue. For example, the representing movement asks how a shared image can be represented and off-loaded so that it can become a manipulable shared space suitable for collaborating. Each worldmaking movement, for example, questions the design of the process whereby public images, forms of life or shared modes of discourse were constructed. Was the design of the movement appropriate? Did the method of inquiry distort the meaning? Is the influencer fragmented because the issue itself is complex or because the movement is inappropriately designed? Was the conversation type appropriate?

The richness of the lens is difficult to portray in a brief overview such as this, so only a few general points can made. Notice that the influencer type in one subsystem usually acts on a similar influencer type in another subsystem. The outer movements act only on appreciators. The inner movements act only on relaters. The expressing movement acts on the generators. The regulator of the conversation subsystem acts on the generator of the language subsystem and is a powerful means of limiting variety within the language subsystem.

The transforming movement, where the regulator of the metaphor subsystem acts on the regulator of the language subsystem, is extremely powerful. People in power are able to impose their metaphors. The domain of acting is governed by language which can be transformed by metaphorically reframing purpose. Thus, a software project manager may redescribe her purpose as being 'conversation

designer' instead of 'engineer'. If this deep metaphor is "lived by" (subsection 3.34), the language of engineering will lose its dominance thus influencing the commitment network to lose its 'production line' mentality. So it can be seen that a change in one part of the system ripples throughout the whole system. In this example, a new image, conversation design, becomes public and reframes purpose, which transforms the language game, which affects the commitment network, which alters activities and forms of life which means, perhaps, that a standard method is seen in a new light and can no longer be legitimised as the standard mode of discourse because the task is not essentially an engineering task, and so on.

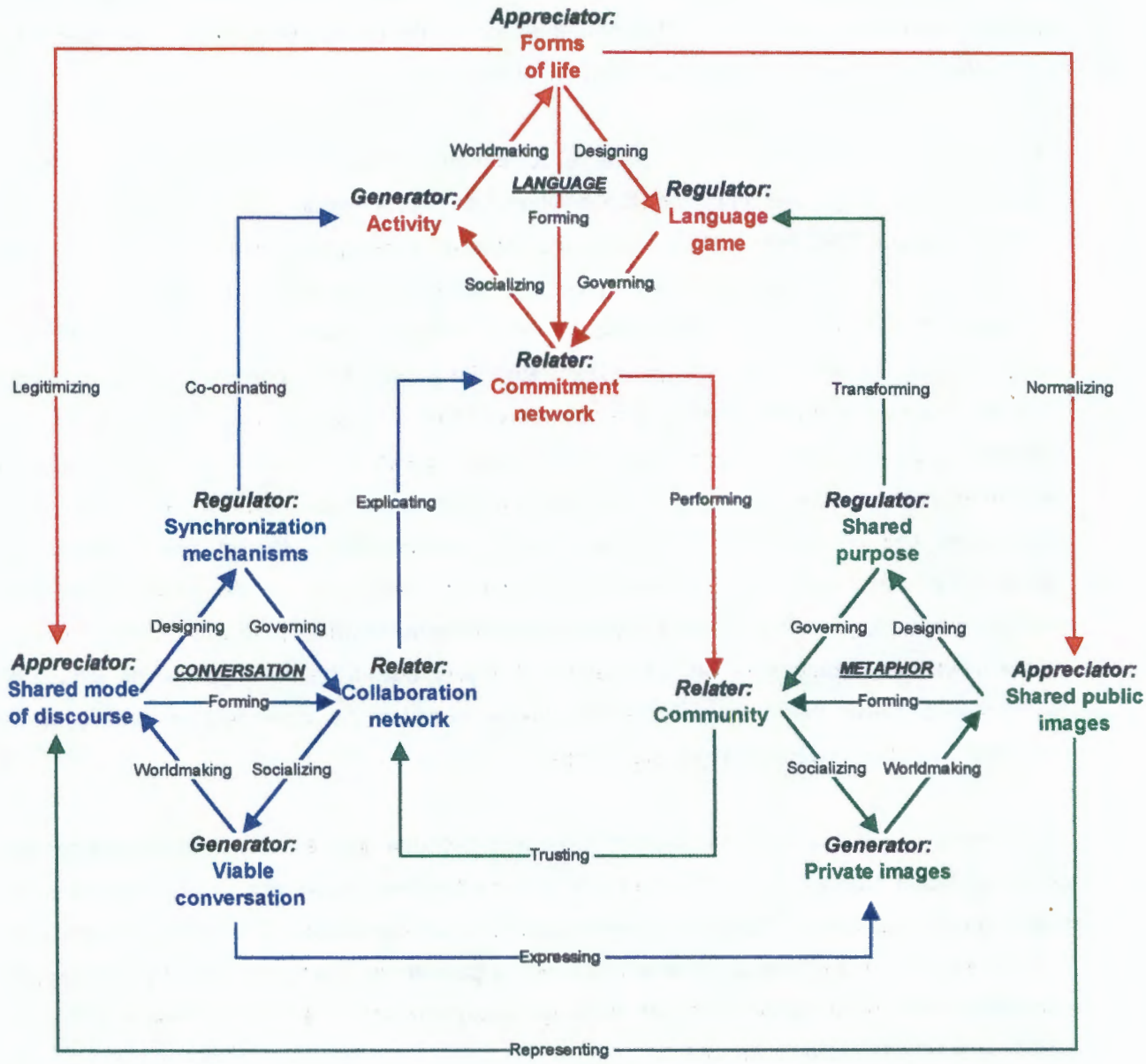


Fig 3. Model for collaborative project design

Some influencers are acted on by only one movement, and others are acted on by as many as three movements. The latter may cause conflicts. For example, shared public images may be historically normalized through a traditional form of life, but recent worldmaking has unfolded new images which conflict with the old. The designer has to contend with a fragmented appreciator, and it is unlikely that a cohesive community will form or "make Polis" around such an issue (Churchman, 1979, page 74).

CPD details social processes explicitly by labelling the movements, but it also embodies the system of influences from Fig 1 and the distinctions and insights categorised in Table 1. CPD thus allows three perspectives on the design of collaborative projects.

1.5 OVERVIEW SUMMARY

This overview has briefly mentioned that the research undertaken in this thesis will attempt to address the software crisis, its associated communication problems, and the need for collaboration during the software process. Chapter 2 will develop an argument that will transform these ideas into a research question and will then outline the research methodology employed to conduct the research. This overview has also outlined the stages in the model building process which will occur in Chapters 3, 4, 5 and 6 of this thesis. The elements of the model building process briefly introduced during this chapter will be grounded, defined and linked in full detail during these subsequent chapters. These chapters will argue that there are useful distinctions to be made between different types of conversation and will relate them to the design of collaborative projects. It will be argued that communication, based on the conduit metaphor, is seldom an appropriate form of conversation for co-ordinating behaviour because it relies on a meaningful, shared interpretative system. It will show that in reality, communication usually requires much more effort than the conduit metaphor suggests because of the need to continually maintain or extend a shared interpretative system. The world re-maker metaphor is put forward as a more realistic metaphor for communication. When there is a critical breakdown in meaningful communication, dialogue is an appropriate type of conversation with which to re-build shared meaning. Dialogue seeks to build meaning by exploring thought itself, and suspending the assumptions of thought for all to see. Collaboration makes use of the meaning generated by dialogue by off-loading collective intelligence into intelligent mind-tools that can be transformed into a shared mode of discourse. In this way, a new project-specific language is created to allow meaningful collaboration in shared space. After the research question has been developed in the next chapter, it will be shown in the remaining chapters that a collaborative project potentially involves the design of three different types of conversation, none of which are well supported by the conduit metaphor.

CHAPTER 2. THE RESEARCH AREA OF CONCERN

2.1 INTRODUCTION

This chapter describes a problem situation affecting software development. There is a persistent productivity problem affecting the process of developing business application software within commercial organisations often referred to as the "Software Crisis" (DeMarco, 1995, page 139). Whilst technological innovation in computer hardware is rapid and fuels great expectations, productivity problems continue to trouble the software process, and the situation is deteriorating. A major concern is that the response to the software crisis by I.T. has been ineffective because it has largely neglected the wider environment which contains the problem, namely the organisational environment. An argument is developed which shows that I.T. is attempting to solve the software crisis on its own and has disempowered the business community on which it depends for software project success. This chapter uses a systems perspective to explore the growing software crisis and to set up a research question to be tested. The fundamental concern is that the software process is systemically inhibited by lack of collaboration in software projects and the research question asks whether this can be alleviated by systemic conversation design. This chapter argues that an engineering metaphor dominates and distorts the software process. Until this can be moderated and enriched by more appropriate metaphors, the software crisis will continue.

Some of the arguments in this chapter are based on 16 years of practical experience in the I.T. industry and on informal experimentation where some of the ideas were already tried out in practice. I feel that this is justified in a chapter that is elaborating on how a research question came into being. Thus not every idea embodied in the research question can be referenced or grounded in a theoretical argument. This research came about precisely because some of these ideas proved to be practically useful and promising enough to require more rigorous research and theoretical grounding. It is especially difficult to argue some of these points from within I.T. literature. Checkland and Holwell agree that much of the conventional wisdom in I.T. textbooks is flawed (Checkland and Holwell, 1998, page xv), that the I.S. industry is confused (ibid., pages 1 - 63), that I.S. literature pays little attention its organisational environment (ibid., page 68) and that I.S generally has an impoverished conception of an organisation which it views as a goal-seeking entity (ibid., page 69). For example, a standard I.T. project management text used by South African technicons makes one reference to an "Organization Model" and this turns out to be the familiar organizational chart which depicts organizational position and reporting lines (McLeod and Smith, 1996, page 67). Similarly, the corresponding systems analysis and design textbook makes a single reference to an organizational model which also turns out to be the same thing (Whitten, Bently and Barlow, 1994, page 230). Because this research project is concerned with the context of software projects, in other words the organisational environment to which I.T. has paid little attention, it is forced to formulate a research

question from conclusions based to some extent on practical experience. It will also draw on systems thinking to bolster these conclusions.

2.2 THE SOFTWARE CRISIS

In 1982, DeMarco states that 15% of software development projects fail completely, and further, that project overruns of 100% to 200% are common. He attributes this mainly to inflated and unreasonable expectations, a symptom of I.T.'s inability to estimate accurately, and to continued reluctance to learn from failures. He feels that failure is usually due to sociology rather than technology (DeMarco, 1995, page 105), and talks of the "high-tech illusion". While some researchers do indeed make high-tech breakthroughs in hardware and software, most developers building business software merely apply these researcher's work, and success or failure depends mainly on human communication skills (DeMarco, 1982).

By 1991 the statistics show no improvement in the productivity or the quality of software development projects. 25% of large software projects never deliver anything, and the average project is a year late and 100% over budget with quality levels of between 1 to 10 errors per 1000 lines of code after delivery to the user. (Yourdon, 1991).

A survey in 1995 by the Standish Group entitled "Chaos" reveals the continuing downward trend:
- 31% of software projects never deliver anything
- only 16.2% are on time and within budget
- average delivery times overrun by 222%
- 52.7% will cost 189% of the original estimates
- only 42% of originally proposed features and functions will be delivered
- for every 100 projects that start, 94 need to be restarted
- the U.S.A. will pay $81 billion in 1995 for cancelled projects
- the U.S.A. will pay $51 billion in 1995 for time overruns (Standish Group, 1995)

The Chaos survey developed from its findings a point count method for assessing the potential for a project to succeed or fail. A count of 100 points indicates a high chance of success:

19	- User Involvement
16	- Executive management support
15	- Clear statement of requirements
11	- Proper planning
10	- Realistic expectations
9	- Smaller project milestones
8	- Competent staff
6	- Ownership
3	- Clear vision & objectives
3	- Hard-working, focused staff
100	- Total points

(Standish Group, 1995)

A 1998 book entitled "Crash" argues that 60% of new computer projects fail (Collins and Bicknell, 1998, page 21). By November 1997, an article in Computing SA shows that the "Chaos" figures have deteriorated so that 40% of I.T application development projects are cancelled before completion, and 33% of the remainder are challenged by cost or time overruns or changes is scope (Field, 1997, page 24). The same article provides the following ten signs of I.S. project failure:

1. Project managers don't understand users' needs
2. Scope is ill-defined
3. Project changes are managed poorly
4. Chosen technology changes
5. Business needs change
6. Deadlines are unrealistic
7. Users are resistant
8. Sponsorship is lost
9. Project lacks people with appropriate skills
10. Best practices and lessons learned are ignored

(Field, 1997, page 24)

It is interesting to note from all the above that technology does not appear to be the main influencing factor in software project failure. A high percentage of errors are introduced into a computer system during requirement analysis, the technology independent phase in which user requirements are specified. These errors tend to be the most difficult to fix.

The software crisis strains relations and inhibits collaboration between the business community and I.T. The business community laments the expense, the delays, the poor quality and often the outright failure of the software process. Their expectations are seldom met. The I.T. community, on the other hand, feel confident about their technical abilities, but laments the lack of proactive business involvement and the inability of the business community to be decisive and certain about system requirements. They feel that they are not given a fair chance to deliver quality. A systems perspective can give insight into this crisis.

2.3 COMPARISON BETWEEN HUMAN POPULATED SYSTEMS AND COMPUTER SYSTEMS

Kauffman lists the following characteristics of complex systems:

1. Self-stabilising - many negative feedback loops
2. Goal-seeking - purposeful, seems to have a mind of its own
3. Program-following - able to follow a program or sequence of steps
4. Self-reprogramming - able to learn, amend its own program, avoid repeating errors
5. Anticipating - has insight into situations through mental models
6. Environment modifying - improves the environment to suit its needs
7. Self-replicating - can reproduce
8. Self-maintaining - can repair itself
9. Self-reorganising - can reorganise relationships between its own parts
10. Self-programming - invents its own goals

(Kauffman, 1980, page 29 to 32)

Human populated systems are complex when considered in relation to these characteristics. Computer systems, especially business application systems, are generally simple by comparison as Table 2 illustrates.

Complex System Characteristics	Human Populated system	Computer System
Self-stabilising	Yes	No
Goal-seeking	Yes	No
Program-following	Yes	Yes
Self-programming	Yes	No
Anticipating	Yes	No
Environment-modifying	Yes	No
Self-replicating	Yes	No
Self-maintaining	Yes	No
Self-reorganising	Yes	No
Self-reprogramming	Yes	No

Table 2. Relative complexity of Human Populated Systems and Computer Systems

Computer systems are perceived to be complex because of their extensive program-following characteristics but, in systems terms, they are otherwise fairly simple systems. They are entirely deterministic, unable to learn, and have no concept of the underlying goals or business principles which they are executing. They are capable of pre-planned responses only and need constant maintenance. New responses require reconstruction of the system. They are constructed from tight definitions and rigid business rules which are cemented together via program logic. Computer systems must be cohesive in order to work successfully and are inflexible by definition.

The business environment, on the other hand, is a complex, human populated system. However hard it may struggle to attain tight cohesion, it is a great deal less cohesive than a computer system. All of Kauffman's ten characteristics are present. Human populated systems are fluid, self-healing and attain dynamic stability far from equilibrium via their capacity to learn. Loose consensus is maintained by diverse patterns of organisational conversation. Compared to a computer system, a human populated system can operate successfully at lower levels of cohesion, less precise definitions, less rigid rules and many differences of opinion. The role of management is to ensure an acceptable level of consensus for the organisation to function. Table 3 summarises some of the differences between human populated systems and computer systems.

In the last row of Table 3, I have introduced the terms "loose complexity" and "cohesive simplicity" as catchall phrases which summarise the nature of the two types of systems. The next section will use these two terms to elaborate on the nature of the software process.

Human Populated System	Computer System
• Goals, purpose	• Instructions, rules
• Rich, fuzzy language	• Limited command set
• Concepts, metaphor	• Algorithms
• Conversation, dialogue	• Programs
• Consensual domain	• Web of logic
• Multiple mental models	• Version control
• Self-healing, learns	• Deterministic
• Human activity	• Simulation of human activity
• Soft unstructured problems	• Hard structured problems, bugs
• Loose complexity	• Cohesive simplicity

Table 3. Comparison between Human Populated Systems and Computer Systems

2.4 THE SOFTWARE PROCESS

Essentially, programming is the process of translating a logically coherent set of rules or concepts into a cohesive set of rigid instructions for a computer to execute. A computer system is a deterministic simulation of a limited part of the business environment which contains it, and is based on a formalization of some aspect of that world (Winograd and Flores, 1986, page 83). Successful programming depends on the ability to represent a subject domain within a computer's internal storage and then to design a set of operations to manipulate this representation (ibid., page 84-85). Typically, a systems analyst attempts to find business rules and user requirements in this environment and then model them using various structured analysis and design techniques. Once the rules are found and logically linked together, it is *comparatively* easy to write business application programs.

The software process is essentially a journey to and from the loose complexity of human populated systems to the cohesive simplicity of computer systems. Software development is characterised by the search for rules, whereas software maintenance is characterised by a search for a rule's context in the business environment. Fig. 4 illustrates this process. The software process revolves around the difficulty of nailing down hard rules, rigid concepts and precise definitions from the relatively uncohesive business environment. Capra illustrates the difficulty in a discussion on expert systems:

"expert systems never perform as well as human experts, who do not operate by applying a sequence of rules, but act on the basis of their intuitive grasp of an entire constellation of facts. ..... expert systems are designed by asking human experts for the relevant rules. When this is done, experts tend to state the rules they remember from the time when they were beginners, but which they stopped using when they became experts. If these rules are programmed into a computer, the resulting expert system will outperform a human beginner using the same rules but can never rival a true expert." (Capra, 1996, page 270)

Reducing loose complexity to cohesive simplicity is a painful and unnatural struggle, and it usually involves complex organisational conversations. The humans who have learned to be expert in the

human system find the precision required by the computer system tedious and unnatural. Experts have to take an unnatural step backward from being unconsciously skilled to consciously skilled as they attempt to remember old facts, elusive rules and lost threads of history. Working concepts become maddeningly difficult to pin down. Imagine, for example, a conversation in which a community of barbers attempt to reach agreement on how to instruct a robot to identify bald men. Does the concept of baldness revolve around a hair count? How many hairs? Are there other ways of determining baldness, like the amount of visible forehead or crown? What is the definition of forehead or crown? This kind of conversation is unnatural for anyone with a working knowledge of baldness, but this is exactly the type of conversation which occurs during systems development. It is often difficult to define even the most basic concepts such as "customer".

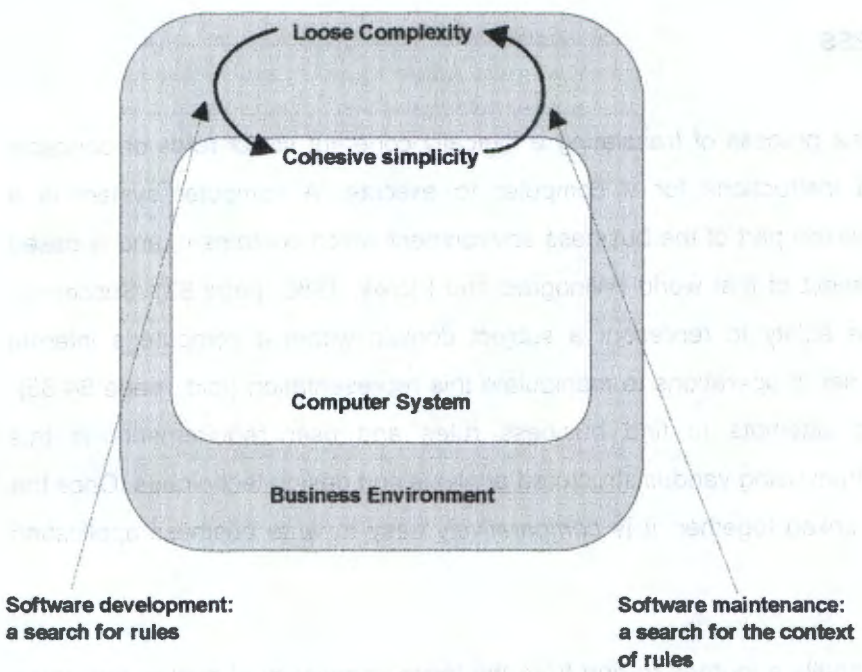


Fig 4. The software process

It will become clear in the next few sections that I.T.'s response to the problem of designing and managing these types of conversation has been minimal, and this contributes greatly to the software crisis. The language of humans is rich. There are words and metaphors for all occasions. In contrast, computer languages have limited command sets of only a few hundred words, each word having one precise meaning. A single word used in the human system may need many lines of complex program code. The challenge in programming is to simulate the richness of the human system with this limited command set. The next section examines how I.T. has responded to this challenge.

2.5 I.T.'S RESPONSE TO THE SOFTWARE CRISIS

The last fifteen years has seen a steady decrease in the success of software development projects, yet there has been progressive adoption of software development tools and techniques during the

same period. It will not be possible to review all of these in the space allowed, but some of the main areas of progress will be highlighted. I will concentrate mainly on software development, as previous sections have suggested that most project failure has little to do with hardware technology.

Fifteen years ago, COBOL was the predominant language of business application systems and there was much emphasis on "structured programming" in an attempt to make programs more understandable and easier to maintain by eliminating "spaghetti code". Fourth generation languages (4GL) became more prevalent in an attempt to make programming quicker. In theory, the syntax of these languages was closer to a program specification language and more conducive to rapid prototyping.

Attention started to shift toward "the front end" of the systems development life cycle, the requirement specification phase, resulting in structured analysis and design techniques. The emphasis was on building logical models of business processes and data in an attempt to build more rigour into the system specification process and to improve communication with the business user. Joint Application Design (JAD) was an attempt to solve the user ownership and involvement problem which has plagued the software development process, and to ensure that systems met user requirements. JAD is an analysis or design workshop attended by end users facilitated by a session leader and is appropriate for data modelling, process modelling, and conceptual systems design. Inevitably, users have differing views of the business, conflicting requirements and disputes over terms, concepts and definitions. JAD provides an interactive forum for these disputes to be settled and the resulting agreements to be modelled.

The ongoing struggle toward rigorous system specification led to the development of comprehensive "life cycle" methodologies of which SSADM is a good example. Enhanced structured analysis and design techniques were packaged into rigorous and detailed check-lists of activities (e.g. Stages, Steps, and Tasks). Each activity was comprehensively documented with detailed instructions on how it should be performed. Some of the more extensive methodologies were contained in sets of books occupying an entire shelf. Organisations often adopted these methodologies in order to get their systems developers to comply to a standard approach to tackling software development. Computer Assisted Software Engineering (CASE) tools evolved to provide automated support for some of these methodologies. Initially the emphasis was on efficient support for business modelling, but later CASE tools evolved to support the entire development life cycle including physical systems design and code generation from business logic stored in encyclopaedias or repositories. Information Engineering (IE) is an example of a comprehensive systems development methodology supported by a number of CASE tools. The whole life cycle from strategic information systems planning and requirements modelling through to code generation in a variety of programming languages is supported.

Some recent and important developments have not been mentioned but they do not fundamentally alter the following very broad summary of the mainstream initiatives:

- Better program coding (Structured programming)
- Quicker program coding (Fourth generation languages)
- More emphasis on "the front end" (Requirement modelling)
- Better modelling of requirements (Structured analysis and design methods)
- Better user involvement (JAD)
- Method convergence (Comprehensive methodologies)
- Software support for the software process (CASE)
- Automation of the software process (CASE, code generators, IE)

Each of these developments represents useful and significant progress, and yet the software crisis continues. The next section provides a reason for this.

2.6 THE ENVIRONMENTAL FALLACY

C. West Churchman lays down a foundation for systems thinking in the following statement:

".....no problem can be solved simply on its own basis. Every problem has an "environment", to which it is inextricably united" (Churchman, 1979, page 5).

We commit the "environmental fallacy" by failing to take the environment of a problem into account. My argument is that the software crisis is caused to a large extent by the I.T. industry taking too little account of its organisational environment. I.T. commits the environmental fallacy by trying to solve the software crisis purely on its own terms. It can be argued that there is a mismatch between the problems inherent in the software process and I.T.'s response to solving them. Table 4 illustrates this.

PROBLEMS	I.T.'S RESPONSE
The journey (from loose complexity to cohesive simplicity)	Focus on the destination (cohesive simplicity)
Creating requirements	Modelling requirements
Synthesis	Analysis
User involvement	JAD
Creating a cohesive system in the business environment	Assumes discovery of a cohesive system in the business environment
Soft issues	Hard issues
Learning from past	Focus on future

Table 4. Mismatch between the problems of the software crisis and I.T.'s response to them.

Fig. 5 illustrates the area in which I.T. has concentrated its efforts. Only JAD makes an attempt to deal with the journey from loose complexity to cohesive simplicity. The other responses are geared toward more efficient programming or the analytic modelling of cohesive simplicity.

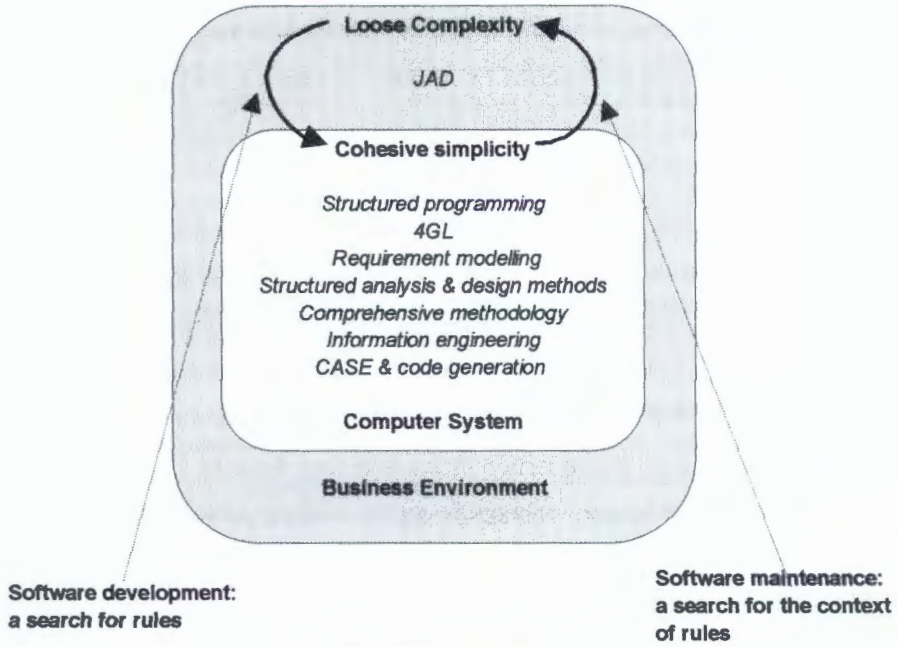


Fig 5. I.T.'s response to the software crisis

It has been shown in section 2.2 that specifying user requirements is a particularly problematic area within the software process. Analysis techniques are well suited to modelling business rules, but offer very little help in bringing business rules into existence. Checkland and Holwell argue that every information system should be conceived of as a pair of systems, a system which is served, and another which does the serving:

"Now, whenever one system serves or supports another, it is a very basic principle of systems thinking that the necessary features of the system which serves can be worked out only on the basis of a *prior* account of the system served. This must be so because the nature of the system served - the way *it* is thought about - will dictate what counts as 'service', and hence what functions the system which provides that service must contain "(Checkland and Holwell, 1998, page 111).

Most I.T. development methodologies assume that there *are* such things as "business requirements", but this is seldom the case. For example, the widely adopted SSADM methodology starts with the assumption that user requirements are already established (Checkland and Holwell, 1998, page 122). In this way, SSADM concerns itself only with the analysis and design of the serving system and remains silent on how to conduct the organisational analysis and design of the system to be served. In general, the need to start with an analysis of the system to be served is not yet part of conventional software process wisdom. There is a misguided assumption that there is a coherent organisational planning system which delivers user specifications (ibid., page 123).

In a similar vein, Weinberg argues that the term "systems analyst" is a misnomer. A more appropriate term would be "systems synthesist" as this emphasizes that systems have to be built into a cohesive whole as opposed to being torn apart through analysis (Weinberg, 1988, page 9). It is difficult to model a system which does not tangibly exist. This is not helped by the fact that the modelling techniques have not proved to be very useful in providing a common language for user and system developer to communicate with each other. The techniques have been developed by I.T. people with an I.T. mindset for I.T. purposes.

There is always the false expectation that the right "technological" solution to the software crisis is just around the corner. This prevents wider inquiry, which should include the perspectives of all stakeholders, into the complexity of developing software in organisations. These hoped-for solutions inhibit learning. Analytic methods continue to be refined at the expense of synthetic approaches, the misunderstandings remain, and I.T. stokes ever-increasing expectations which it fails to deliver. The business community is thus disempowered and prevented from learning how to be proactively involved. The following sections argue that language, metaphor and conversation have much to do with this.

2.7 METAPHOR AND THE SOFTWARE PROCESS

Organisations can be viewed metaphorically (Morgan, 1997). An organisation can be viewed as an on-going conversation (Winograd and Flores, 1986, page 157 to 158). Organisations and divisions within organisations have, in Wittgenstein's term, their own peculiar "language games" (Wittgenstein, 1958, paragraph 7). Those who create the organisation's dominant language tend to control its power. Lakoff and Johnson argue that our conceptual systems are fundamentally metaphoric in nature, and that we tend to structure our activities on the basis of our metaphorically structured conceptual systems (Lakoff and Johnson, 1980, pages 3 to 6). The dominant metaphors within the organisation shape the mental models, language games, and determine the nature of activity.

My argument is that the business community has been disempowered by allowing the metaphors and analogies of I.T. to dominate the software process unnecessarily. Business users and software developers alike tend to use "hard" analogies when describing the software development process. They talk of 'information engineering', 'software engineers', and 'code construction'. Developing software is seen as a technical activity such as building a skyscraper or a space rocket. However, the software crisis is not caused primarily by hard technological problems, and therefore the engineering metaphor needs to be enriched or replaced. For example, software development could be described as being more like compiling a dictionary, like staging a theatrical production, like making a movie, like organisational therapy or like instructing a Martian how to live on planet earth. Mental models of

the software process need to be richer, more accurate, more comprehensive and shared by all stakeholders.

Here is an example of how an analogy may help to uncover the sub-systems which need to be designed and managed. A legal analogy for software development might turn up three sub-systems: law-making, law-enforcement, and law-remembering. The process of turning the needs of the business community into a cohesive set of rigid rules is a type of organisational law-making. Legislation hinges on the precision of words and concepts, and also what is good for society as a whole. Law-enforcement occurs when business rules are coded into computer systems, for a computer system constrains the business system which contains it. This law-enforcement should also cover the environment modifying laws which must ensure that the human system adapts to the computer system, that the human system and the computer system are designed to relate to each other, and that this relationship is able to be tested and enforced. While computer code perfectly documents the letter of the law, albeit in a very obscure way, it does not document the spirit of the law. Some form of remembering system is needed to help people understand why the law has developed the way it has. This could include underlying goals and principles, the reasons why some decisions were made in preference to others, and a history of on-going system evolution. Traditionally, documentation of computer systems has been weak and is normally the first thing to be abandoned when a project starts falling behind schedule. But this is short-sighted, as the average I.T. department spends much of its budget on systems maintenance. Efficient maintenance of a computer system depends on the contextual knowledge which both users and programmers have of the program code. When this context is forgotten, the computer system can take control of the business. The law is followed blindly. Testing a complex computer system which has many interfaces to other systems is almost impossible when there is no overall contextual knowledge.

The point of this is that the engineering metaphor disempowers the business community because it portrays the software process as a foreign technical activity which is therefore best left to I.T. experts. A legal metaphor, on the other hand, emphasises the system to be served and thus empowers the business community by portraying the software process as an activity focused on designing a system of organisational rules. The serving system follows from the design of the system to be served - designing the latter is not fundamentally a technical activity.

2.8 SOFTWARE DEVELOPMENT AS ORGANISATIONAL CONVERSATION

Software development may be viewed as a process of turning conversations into program code. Under the influence of the engineering metaphor, project managers usually act as if they are managing a software production line which takes in user requirements as its raw material and produces program code as its output. The universal complaint that 'users don't know what they want' is a complaint about the raw material not being in a suitable state to allow the production line to

function. Project managers might enable their business partners to become more proactively involved if they present their task as one of designing and managing a complex conversation in which the organisation tries to understand itself and its needs at an unnaturally precise level of cohesion. JAD represents I.T.'s attempt to design such conversations, but ultimately it fails because it uses analysis techniques in a conversation that requires synthesis.

The inappropriate mental model created by the engineering/production line metaphor often results in the entire process of software development being delegated to I.T. along with higher levels of expectation than can actually be coped with. The business stakeholders forgo the up-front ownership role necessary for project success and are reduced to demanding helplessly that delivery deadlines are met. When projects get into trouble, a likely cause is a runaway conversation which results from lack of collaboration. Programmers spend their time fruitlessly seeking direction from business users who in turn think that it is the programmers who should be the experts.

Hard project management needs to be moderated and enriched by viable conversation design. This underlines the point made earlier about the "high tech illusion" and software project failure being caused by sociology rather than by technology (DeMarco, 1982).

2.9 TRANSFORMING THE I.T. LANGUAGE GAME

The systems approach (Churchman, 1979) attempts to provide a framework for inquiry into complex situations. It is especially well suited to the kind of soft, unstructured problems encountered in human populated systems. It sweeps in the multiple perspectives of critically considered stakeholders and instigates a learning process to promote shared understanding of mental models. It has comprehensiveness as its goal and examines the underlying circular causality of the system dynamic so that interventions can be made (Senge, 1994, page 113).

My argument is that the software process is hampered by the language game associated with it. The engineering metaphor promotes a language game which ensures I.T. control. This disempowers the business community and encourages *all* stakeholders to hold false mental models. A suggested "high leverage intervention" (Senge, 1994, page 165) would be the attempt to change the language game through metaphoric reframing (Schon, 1993, 150) of the software process. The objective would be to build shared software process insight and a richer language game so that all stakeholders are appropriately empowered to take part in the project conversation. Only in this way will it be possible to build viable project communities that include the business community as actively involved participants rather than as helpless onlookers. It is possible that deeper insight will encourage the business community to cultivate the business environment by reducing its loose complexity so that it appears more cohesive to software developers. It could also initiate development of new methods of synthesis that deal with that part of the software process which the I.T. industry has largely neglected, namely

loose complexity. Fig 10 in section 4.1 summarises the concerns and insights which influenced the development of the research questions.

2.10 THE RESEARCH QUESTION

Having outlined the research problem and area of concern, an explicit research question can now be developed in this section. A chief concern is that the journey from loose complexity to cohesive simplicity will never be successfully accomplished through the kind of analytical methods which I.T. continues to develop. Similarly, business involvement will remain poor if the software process continues to be dominated by a disempowering metaphor and its concomitant language game. The question to be answered is whether it is possible to develop skills or methods that enable genuine collaboration among *all* stakeholders. Collaboration should be based on appropriate foundations, unlike JAD which attempts to achieve systems synthesis but is based on analytic methods. How is it possible to design collaborative conversations that enable systems synthesis inherent in the journey from loose complexity to cohesive simplicity? This question enlarges the scope of the research into a broader question that extends beyond I.T and the software process. It must be assumed from the evidence of increasing software failure that the business community also does not generally possess such collaborative expertise, otherwise it would have already been used to transform the software process into a collaborative activity. An ability to design collaborative projects is thus a general need.

This chapter has highlighted a cluster of topics. It has emphasized the usefulness of the systems approach and it has argued that language, metaphor and conversation are significant elements affecting the software process. When these elements are combined with a desire to design collaborative projects, we end up with the fundamental research question: is it possible to combine insights from language, metaphor and conversation with the systems approach to enable the design of collaborative projects? This question could be rephrased as follows: is it possible to attain a clear understanding of an organisation by combining systems thinking with perspectives from language, metaphor and conversation? If the answer to the latter question is affirmative, the next question is: how can this be used effectively to design collaborative projects?

The Preface mentions prior practice that inspired this research. This practice was based on 'gut-feel' rather than sound theoretical underpinnings. An additional question related to the research question posed above is whether it is possible to re-interpret the apparent success of this prior practice through a theoretical position based on collaborative project design and supported by combined theoretical perspectives from language, metaphor and conversation and the systems approach.

The research problem, area of concern and area of application remain I.T. and the software process. The research question focuses on the environment of I.T. and seeks to build skills in this environment to complement I.T. skills. Fig 6 depicts a more appropriate response to the software crisis.

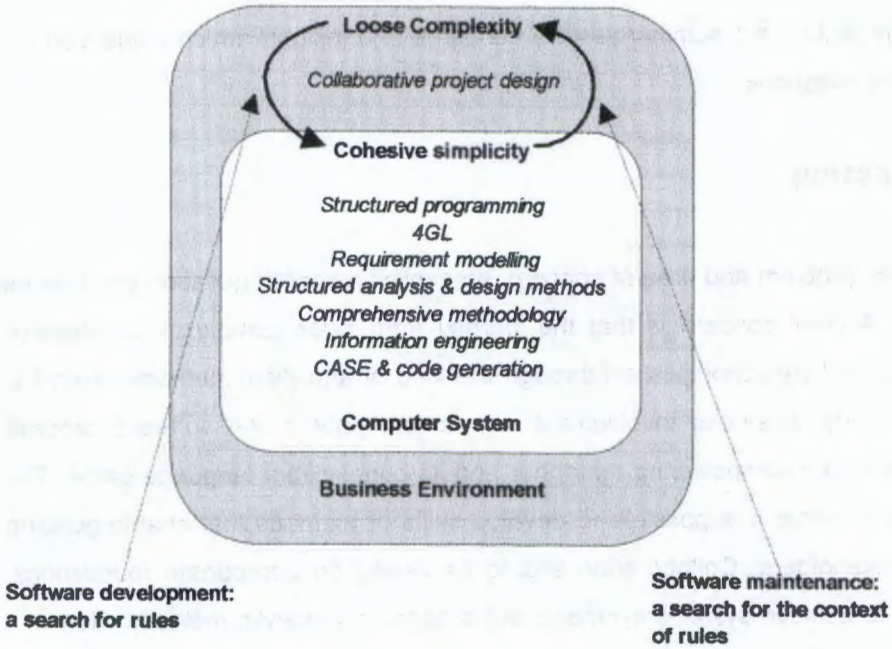


Fig 6. An appropriate response to the software crisis

2.11 THE RESEARCH METHOD

The research questions posed in the previous section were answered by action research. I have argued that the area of concern is characterised by social rather than technical problems, and therefore an interpretative research method is more appropriate than a positivistic method of natural science. The interpretive research model promotes a process of critical inquiry involving a collaboration between researchers and the people in the situation. Here the focus is on social practice and deliberate reflective learning (Argyris et al, 1982). Checkland and Holwell argue that action research is an appropriate method for exploring information provision and the issues surrounding it (Checkland and Holwell, 1998, page 26 - 27). They point out the difference between positivistic and interpretive methods:

"The implicit belief behind hypothesis-testing research in information systems is that social phenomena and social reality are at core not fundamentally different from the physical reality which biologists, chemists and physicists investigate. An alternative view is that social reality - what counts as 'fact' about the social world - is continually being constructed and re-constructed in dialogue and discourse among human beings, and in action which they take. Researching social reality then becomes an organized discovery of how human agents *make sense* of their perceived worlds, and how those perceptions change over time and differ from one person or group to another. This kind of researcher does not expect to find unchanging 'social laws' to set alongside the laws of physics." (ibid., page 22)

Kurt Lewin is normally credited as initiating the development of action research as a method of researching human situations outside laboratory situations (ibid, page 22). His method for conducting action research can be summarised as follows:

1. Planning starts with a general idea which gives rise to the desirability of achieving a certain objective. The idea is carefully examined in light of the means available.
2. Frequently more fact-finding about the situation is required.

3. An overall plan emerges on how to reach the objective accompanied by a decision with regard to the first step of action.
4. The first step of the overall plan is executed.
5. There is a reconnaissance or fact-finding which has four functions:
    5.1 The action is evaluated to assess whether its achievements are in line with expectations
    5.2 It give the planners a chance to learn
    5.3 Additional fact-finding serves as a basis for planning the next step
    5.4 It serves as a basis for modifying the overall plan
6. The next step again is composed of a circle of planning, executing and reconnaissance for the purpose of evaluating the second step, preparing the rational basis for planning the third step and for perhaps modifying the overall plan. (Lewin, 1946, page 145-146)

Lewin thus argues that:

"Rational social management, therefore proceeds in a spiral of steps each of which is composed of a circle of planning, action and fact-finding about the result of the action" (ibid., page 146)

The following action research process is put forward as a guide for practitioners operating in organisations:
1. Identify the problem or difficulty
2. Suggest proposals for action development and action planning
3. Select action steps and formulate a hypothesis for testing
4. Plan data collection
5. Gather data or evidence
6. Draw conclusions
7. Communicate findings or solutions
8. Emphasise learning from action
(Bennett and Oliver, 1988, page 3 - 4)

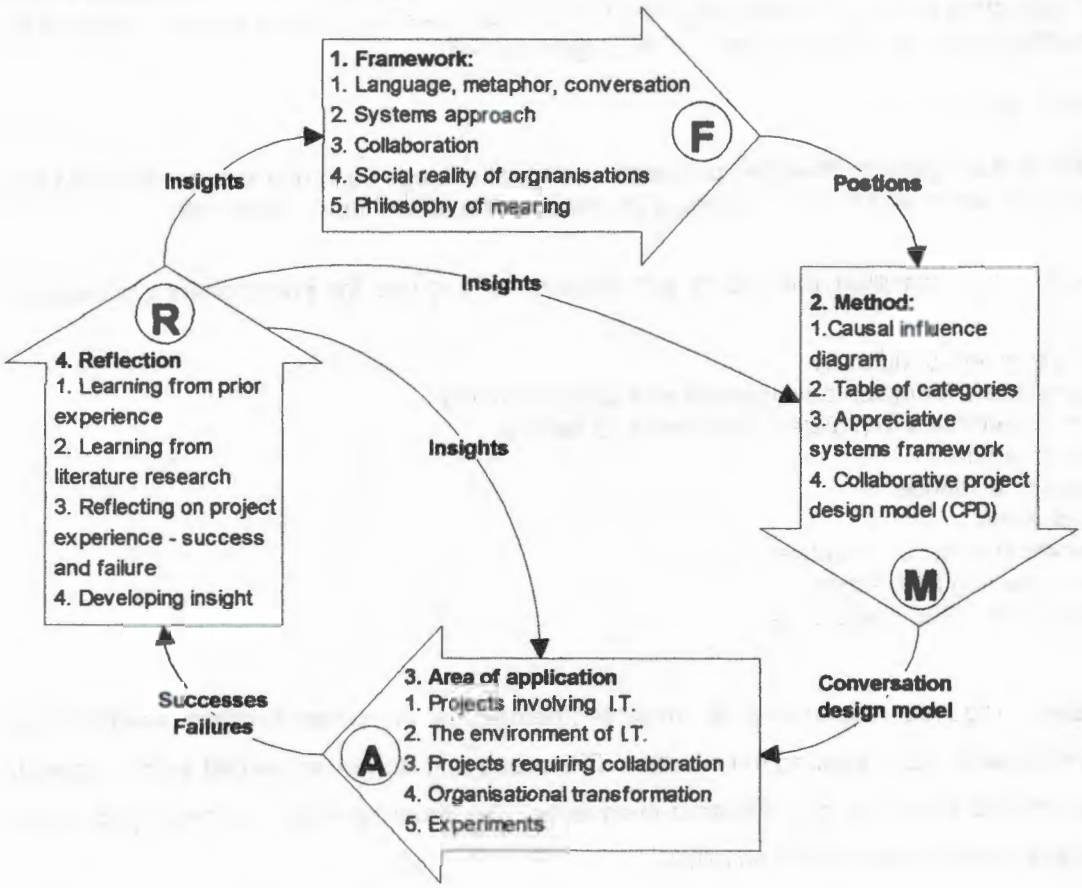
Action research requires researchers to immerse themselves in human problem situations as participants contributing to improving the situation. Checkland and Holwell argue that action research should be conducted according to "a declared-in-advance intellectual framework of ideas" (ibid., page 22). Their model of action research is as follows:

1. There is one or more research themes.
2. The researcher declares in advance a framework (F) of ideas and embodies them in a methodology (M) to be applied in an area of concern (A).
3. The researcher enters a real world problem situation (A) and takes an active part in the change process.
4. The researcher reflects on experience and involvement in the situation.
5. Reflection may lead to learning in relation to F, M or A.
6. New findings can be incorporated into new or changed F, M or A. New research themes may be developed.
7. This cycle may be iterated a number of times. (ibid., page 26)

A hypothesis testing approach such as Bennett and Oliver's was rejected for its positivistic overtones. Checkland and Holwells model for action research was chosen for a number reasons. Their insistence of adding a "declared-in-advance epistemological framework" (ibid, page 23) model adds a crucial missing element to the models of Lewin and Argyris et al. The argue:

"Without a declared-in-advance epistemological framework it is sometimes difficult to distinguish researching from novel writing. Such a declared framework also allows those interested in the research and its outcomes to recover the process by which the results were obtained" (Ibid., page 23)

In addition to this, Checkland's approach to action research has spawned SSM which is the systems thinking methodology which has had most impact on the software process. For example, Multiview is a software methodolgy which is based on SSM (Wood-Harper et al, 1985). The FOAR methodology (Schafer, 1988) and the COMPACT approach (in Patching, 1990, page 138) are further examples.



Schema 2 - The research methodology

Schema 2 shows that the action research conducted during this research project was very similar to the research model of Checkland and Holwell. The research process began with concerns and questions about a problem area, in this case the software process. Informal research, practitioner's experience, prior insights, and prior successes with these insights all combined to provide a basic, initial framework of ideas. Step 1 shows that the framework was built from a cluster of ideas that were developed into positions to be embodied into a method. Step 2 created a model to be applied in real world projects. Step 3 used the model with differing degrees of success or failure. Successes and failures both provided opportunities for reflection. Step 4 generated insights from the results of action research and also from literature research. These insights enriched or altered the framework of ideas or the method, and provided new understandings about the area of concern. The research process iterated until the framework, the positions, the method, the model, and the insights were routinely useful in improving the initial problem area of concern.

Step 1 is concerned with positions to be incorporated into the method. Points 1, 2, and 3 of the framework of ideas (F) have already been developed in this chapter. The research question states the wish to combine language, metaphor and conversation with the systems approach in an attempt to find ways of designing collaborative projects. The need to understand the organisational environment prompted early readings to focus on philosophy. Within the first six months of the research process it became apparent that the social construction of reality (point 4) combined with a philosophy of meaning (point 5) completed the framework of ideas. This framework was steadily explored and increasingly refined, but not fundamentally changed throughout the research process.

Causal influence diagrams similar to those popularised by Senge (Senge, 1990, page 68-92) and Kauffman (Kauffman, 1980, page 6-28) were the major tool for organising the research process. Fig 1 represents the final model which was settled on after a great many versions had been considered. Early literature research focused on attempting to understand organisations, and this modelling approach was important as a means of consolidating the learning. It was important to get a 'picture' that was grounded on solid philosophical foundations about how organisations worked. The philosophy of meaning and social construction of reality provided the necessary positions on the nature of organisations and causal influence diagrams provided a useful mode of representing this learning. It was especially important to be able to represent and understand organisations as systems of causal influences. The causal influence diagrams became an empowering lens through which to view the social reality of collaborative projects. Early projects were concerned with validating, enriching or refining the causal influence diagram. It is important to note that practical work was being conducted throughout the whole research period so that literature research and practical research were being conducted simultaneously. The causal influence diagram was a helpful tool to keep theoretical and practical research synchronised and recorded.

In the initial stages, the plan was to build an explicit methodology for the design of collaborative projects. This would have entailed explicit tasks, steps, facilitation procedures and so on. But it became apparent that having a model of the social reality of collaborative projects provided an empowering lens which enabled the design of appropriate interventions. Instead of switching from the causal influence diagram to a methodology based on its findings, I decided to enrich the causal influence diagram itself to make it into a more useful lens. The outcome of this research is *not* a methodology, but a systemic model that depicts the dynamics of the social reality of collaborative projects. The purpose of this model is to allow the project situation to be clearly 'seen' so that appropriate conversations can be designed to improve collaboration. In order for the model or lens to be rich enough to do this, a table of categories and distinctions was developed. The projects themselves will each be described in Chapter 7, but essentially they all involve collaboration in projects directly or indirectly involving I.T in financial services organisations. The research process thus progressed as follows:

Having justified Checkland and Holwell's model for action research and introduced the elements of the research method, it is now possible to make the research methodology explicit:

1. A declared-in-advance epistemological framework is synthesised from prior practice, literature research and previous action research.
2. This synthesised framework is consolidated in a type of systems model similar to Fig 1. The systems model becomes the mental construct with which to organise the mind of the practitioner when intervening in a problem situation.
3. The method of research is as follows. The practitioner's mental model is used to as the lens with which to "see" the problem situation. "Seeing" can involve observing activity, making inquiry and appreciating the problem situation in terms of the systemic mental model. The practitioner will seek out from the total set of influences in the real problem situation that subset of influences which correspond to the systemic mental model.
4. Having identified the state of the relevant subset of influences in the real world, the practitioner can design interventions to modify the state of influences in the real world. This design of interventions is guided by the mental system of influences.
5. There is a learning process which assess the effectiveness of the intervention. A failed intervention would result in possible revision of either the declared-in-advance framework or the systems model.
6. The cycle for 1 to 5 is repeated so that the frameworks, models and method are steadily enriched. Early models consisted of a relatively rudimentary influence diagram (i.e. Fig 10 on page 92) and evolved to include Fig 1, Fig 2 and Fig 3. Table 1 also evolved through this action research method and became the vehicle for consolidating an increasingly richer declared-in-advance epistemological framework. It is thus vital to understand that the elements of CPD (Fig 1, Fig 2, Fig 3 and Table 1) are not only the output of the research, but they are also the vehicles for the research.

It must be acknowledged that, although literature research and practical research were conducted simultaneously, practice tended to precede theory. I was not always able to control the commencement dates of the research projects because the arrangements that I had made for conducting research had to adapt to the ongoing projects, contracts and crises of various organisations. Because I had been experimenting fairly successfully with conversation design for a number of years prior to deciding to do formal research and had thereby earned some confidence from various colleagues, I tended to 'deal with' the projects as they presented themselves. For example, Projects 1 and 2 detailed in chapter 7 were thrust upon me before there was time to finalise any theoretical basis for my insights. I decided to learn as much as I could from practical situations as the opportunity arose. In retrospect I feel that my insight from theoretical readings was enhanced by my simultaneous practical research.

CHAPTER 3. CATEGORIES FOR THE DESIGN OF COLLABORATIVE PROJECTS

3.1 INTRODUCTION

Chapter 1 of this thesis gave a brief overview of the derivation of a model for collaborative project design (CPD) and Chapter 2 motivated the need for such a model as a contribution to solving the software crisis. An important step in deriving CPD was to build a table of categories for the three domains of collaborative projects, namely acting, thinking and conversing (Chapter 1, page 12, has already described the structure and category types of Table 1). Section 1.1 listed these categorizations and provides an explicit mapping of elements in Table 1 to the terms introduced in this chapter, but was not able to argue in full why each categorization was included in the table nor its full relevance to collaborative projects. This chapter rectifies this by motivating each categorization within each domain. The headings of each section in this chapter correspond to a domain within Table 1 and each subsection heading corresponds to specific categorizations within that domain. Thus, it will be necessary for the reader to continually align the subsection headings with the categorizations of Table 1 in order to fully appreciate the direction of the argument. The fold-outs of Schema 1 and Table 1 should help to orientate the reader and assist in assimilating the contents of this chapter. In particular, this chapter corresponds to Step 1 of Schema 1.

3.2 ACTING

This section will draw from the philosophies of Wittgenstein and Heidegger. Winograd and Flores have drawn heavily on the latter in arguing that an organisation can be viewed as a conversation focused on maintaining a network of commitments. Because they were also concerned with cognition and computers, their interpretation of Heidegger is of great relevance to the practitioner struggling with collaborative projects and the software crisis.

Finch, who considers Wittgenstein and Heidegger to be the two foremost philosophers of the age, points out that they both were forced to abandon their attempted search for *absolute grounds* to support their initial philosophies of understanding and interpretation (Heidegger), and sense and reference (Wittgenstein) (Finch, 1995, page 12-13). In finding their way independently to *post-metaphysical philosophies of meaning* they brought about a new direction in philosophy in which epistemology becomes redundant. This change from *knowing* the meaning to *doing* the meaning emphasizes activity. They reject the notion that there is a human knowing subject who confronts an objective world and knows it primarily through mental representations. (ibid., page 10-13)

3.2.1 Language, activities, forms of Life, and language games

Wittgenstein believes that our way of life is mirrored in our language and that it is a mistake to try to understand language in isolation from the context in which it normally does its work. "To imagine a language means to imagine a form of life" (ibid., paragraph 19). A form of life is an activity or collection of related activities of a certain kind which become regularized. Language is woven into human activities and behaviour, and gets its content from our practical affairs. "Speaking a language is part of an activity, or a form of life."(ibid., paragraph 23). Language is a learned substitute for more primitive or natural expressions. Crying, for example, is a natural expression of pain behaviour, whereas saying "ouch" is a culturally learned substitute for pain behaviour. "Commanding, questioning, recounting, chatting, are as much a part of our natural history as walking, eating, drinking, playing" (Wittgenstein, 1958, paragraph 25). The difference between verbal behaviour and other behaviour is therefore one of degree, not of kind.

Wittgenstein's root unit of meaning comprises three interrelated elements: human beings, a world-setting, and language (Finch, 1995, page 44). This whole, consisting of language and the actions into which it is woven, he calls a "language-game" (Wittgenstein, 1958, paragraph 7). Wittgenstein sees language as a collection of language games which can be used to describe, report, inform, affirm, deny, speculate, give orders, ask questions, tell stories etc. The 'game' metaphor stems from his observation that there is no single criterion which is common to all games, although through a "complicated network of similarities overlapping and criss-crossing" they bear a "family resemblance" to each other (ibid., paragraph 66 - 67). A form of life gives full sense to language activity, and while it does not explain the language game, it does make it comprehensible. Finch gives an example which helps to explain the relationship between language games and forms of life (Finch, 1995, page 51):

language games	forms of life
advertising	buying and selling (commerce)
psychoanalysis	therapy (dream interpretation)
haruspicating	divination
betting on lotteries	gambling
exorcism	magic ritual

Table 5. Examples of language games and forms of life

Unlike language games, forms of life may include each other, and are usable at different levels of generality. Thus the human form of life includes other forms of life such as war, business, education and religion (ibid., page 53).

The meaning of an expression is determined by its use in a language game. It is fixed by publicly agreed rules based on a shared form of life. Grayling summarizes Wittgenstein's account of meaning as follows:

"The meaning of an expression is what we understand when we understand that expression. Understanding consists in knowing the expression's use across the variety of language-games in which it occurs. Knowing its use is having an ability: the ability to follow the rules for its use in those different language-games. Rule-following is not a mysterious inner process of grasping something like a calculus which objectively imposes standards of correctness; rather, it is a practice embedded in the customs and agreements of a community and as such is essentially public. Rules do indeed guide and provide standards of correctness - but they do so because they are based on agreement; to follow a rule correctly is to conform to the established practices of the community. We acquire the ability to use expressions - to follow the rules for their use - by our training as members of that community. (Grayling, 1988, page 83)

Meaning is neither a denoting relationship between words and things, nor a picturing relationship between words and facts. Understanding the meaning of expressions does not lie in private processes or mental representations. Language games cannot be grounded or justified or explained. They can only be used and described. They cannot be grounded with absolute certainty according to external criteria. The only certainty available is the ability of the language game to "stand fast" (Wittgenstein, 1969, paragraph 116) as a coherent context for meaningful activity among a community of language users.

3.2.2 Background - taken for granted and ready-to-hand, no mental representations, individual expert

Winograd and Flores reject the dualism of the rationalistic tradition which posits an objective world of physical reality and a subjective mental world of the individual mediated by language as a system of symbols that stand for things in the world. They reject cognition as the systematic manipulation of representations of an objective world. Because meanings of items can not be fixed without reference to the context in which they appear, they emphasize instead interpretation and the role of the observer or listener in the active generation of meaning. Interpretation pervades everyday life. Language is learned through interpretation but it is through language with its implicit assumptions, prejudices and pre-understandings that we interpret phenomena and generate meaning. Thus, "existence is interpretation and interpretation is existence" (Winograd and Flores, 1986, page 31).

Winograd and Flores highlight four main points which represent Heidegger's philosophy:

1. Our implicit beliefs and assumptions cannot all be made explicit.
2. Practical understanding is more fundamental than detached theoretical understanding. We have primary access to the world through practical involvement with ready-to-hand - the world in which we act unreflectingly. Detached contemplation can be illuminating but it also obscures the phenomena themselves by isolating and categorizing them.
3. We do not relate to things primarily through having representations of them. The focus is on concernful activity instead of detached contemplation. My ability to hammer comes from my familiarity with *hammering*, not my knowledge of a *hammer*.
4. Meaning is fundamentally social. (ibid., page 32 - 33)

We cannot avoid acting and we are in effect *thrown* into action against our wills. Interacting with other people puts us in a position of *thrownness*. While we are acting as individuals with expertise, we cannot step back and reflect on our actions, because we need to respond immediately. We are *thrown* on our instincts to deal with whatever comes up. The effects of our actions cannot be predicted and we do not have a stable representation of a situation while we are in it. A representation might be developed from an analysis of the situation after the event, but this was not the understanding we had as the situation was developing. Every representation is an interpretation, not an objective analysis. Each time we speak in a situation we are doing something quite different from simply 'stating a fact'. We are not describing the situation but creating it. In mentioning something, we bring our interpretations into the social discourse and we create the objects and properties we describe by virtue of making an utterance. Thus, language is action. (ibid., page 34-35)

The background world within which we act is taken for granted. Objects and properties are not explicitly recognised nor identified and are not consciously *present* in the situation but are unconsciously *ready-to-hand*. There is a familiar ability to act without reflection or conscious knowledge. For the person engaged in the thrownness of unhampered hammering, a hammer does not present itself as an entity any more than the tendons of the hammerer's arm. Objects and properties are not inherent in the world, but emerge only in the event of a *breaking down* in which they become *present-to-hand*. A torn tendon would present itself in the breakdown of hammering as something that needed to be consciously reflected upon, coped with and probably discussed (ibid. page 36). It is meaningless to talk about the existence of objects and their properties in the absence of concernful activity with its potential for breaking down. (ibid., page 37)

3.2.3 Observation and description

Because we can never have knowledge about external reality, it is naïve to view language as something that conveys information about objective reality (ibid., page 50). We can talk about the world but in doing so we act as observers who generate distinctions which cannot be grounded in external reality, but only in the consensual domain of a community of observers (ibid., page 50 - 51). All observations and descriptions necessarily introduce blindness because they highlight certain distinctions and promote "a readiness to see" (Vickers, 1968, page 41) things in terms of those distinctions. This necessarily promotes a blindness to categorisations and distinctions that were not made.

3.2.4 Representation paradox

This topic is introduced now, but will be discussed further in subsection 3.4.8. An expert experiencing the thrownness of unreflecting, concernful activity where everything is ready-to-hand does not have a mental representation or description of the activity that is being performed. This provides a major challenge for any systems designer who is relying on enlightenment from an expert. The observer is driven to make representations and distinctions from a particular perspective that the expert does not need and which will probably puzzle the expert. But the expert with no need for representations is in no position to help the 'outside' observer and may have no ready-made mode of representation with which to satisfy the need of the observer. This paradox could be called a 'crisis of representation' as it so often lies at the heart of collaboration problems. Who determines the appropriate mode of representation and on what basis? Is it the observer with no 'insight' or the expert with no 'outsight'? When the motivation and basis underlying the chosen mode of representation are not understood by one party, collaboration deteriorates. The conversation is no longer an act of shared creation or discovery, but a one-way question-and-answer session controlled by one party.

3.2.5 The "submarine analogy"

Maturana and Varela give a good example of the representation paradox:

"Imagine a person who has always lived in a submarine. He has never left it and has been trained how to handle it. Now, we are standing on the shore and see the submarine gracefully surfacing. We then get on the radio and tell the navigator inside: "Congratulations! You avoided the reefs and surfaced beautifully. You really know how to handle a submarine." The navigator in the submarine, however, is perplexed: "What's this about reefs and surfacing? All I did was push some levers and turn knobs and make certain relationships between indicators as I operated levers and knobs. It was all done in a prescribed sequence which I'm used to. I didn't do any special maneuver, and on top of that, you talk to me about a submarine. You must be kidding!"

All that exists for the man inside the submarine are indicator readings, their transitions and ways of obtaining specific relations between them. It is only for us on the outside, who see how relations change between the submarine and its environment, that the submarine's

behaviour exists and that it appears more or less adequate according to the consequences involved. ....The dynamics of the submarine's different states, with its navigator who does not know the outside world, never occurs in an operation with representations of the world that the outside observer sees..." (Maturana and Varela, 1987, page 136 - 137)

3.2.6 Speech acts and commitment networks

Linguistic forms are shaped by the need for effective coordination of action with others. Activity is coordinated by speech acts that bind speakers and listeners in mutual commitments. Speech acts take place against a background of obviousness shared between speaker and listener. If utterances are not intelligible or interpretations between speaker and listener are not consistent there will be breakdown resulting in the loss of mutual trust in commitment. (Winograd and Flores, 1986, page 62-63) Here is a brief summary of Searle's taxonomy of illocutionary acts:

1. Assertives commit the speaker to something being the case.
2. Directives attempt to get the hearer to do something.
3. Commissives commit the speaker to future course of action.
4. Expressives express a psychological state about a state affairs.
5. Declarations bring about a correspondence between the speech act and reality.

(ibid., page 58-59)

Conversations come about when successive speech acts are related to one another and can be viewed as a sort of dance involving requests, acceptances, rejections, counter offers, withdrawals, modifications, and so on. At each point in the conversation there is a small set of possible linguistic actions determined by previous history. This interplay of speech acts is directed towards explicit cooperative action which will be performed outside the conversation. (Winograd and Flores, 1986, page 64). When we are engaged in successful conversation, it is not present-at-hand, as something observed. We are immersed or thrown in its unfolding. Meaning arises in listening to the commitment expressed in speech acts. Its structure becomes visible only when there is some kind of breakdown in which case we are committed to provide some kind of explanation or 'grounding'. (Winograd and Flores, 1986, page 68). Communicative competence is dependent on the capacity to effectively express one's intentions and the ability to take responsibilities in the networks of commitments that utterances and interpretations bring to the world (ibid., page 162).

Winograd and Flores take the following position regarding management and organisational conversation:

"1. Organizations exist as networks of directives and commissives. Directives include orders, requests, consultations, and offers; commissives include promises, acceptances and rejections.
2. Breakdowns will inevitably occur and the organization needs to be prepared. In coping with breakdowns, further networks of directives and commissives are generated.
3. People in an organization (including, but not limited to managers) issue utterances, by speaking or writing, to develop conversations required in the organizational network. They participate in the creation and maintenance of a process of communication. At the core of this process is the performance of linguistic acts that bring forth different kinds of commitments.

In fulfilling an organization's external commitments, its personnel are involved in a network of conversations. This network includes requests and promises to fulfill commitments, reports on the conditions of fulfillment, of commitments, reports on external circumstances, declarations of new policies, and so on. The organization encounters requests and other external contingencies that it can deal with by making commitments that can be fulfilled by activation of certain special networks of recurrent conversations, where only certain details of the content of the conversations differ, not their general structure. These networks of conversations are at the core of the organization. They are embodied as intercommunicating offices, each specialized in fulfilling certain kinds of commitment.

A person working within an organization is always concerned with questions such as "What is missing?", "What needs to be done?", and "Where do I stand in terms of my obligations and opportunities?" In situations where many people must act together, the problem of co-ordination becomes crucial. For many organizations it is a matter of survival. " (ibid., page 157)

The philosophy of meaning emphasises concernful activity in which an individual acts unreflectingly within a taken-for-granted background. From the designer's perspective, the world of acting is focused on performance relative to expectations inherent in the commitments that project members make to each other through speech acts. Success revolves around the ability to coordinate the networks of commitments through a process of communication made meaningful by shared forms of life and governed by a coherent language game. It is only in breakdown that the objects and properties of the world emerge, become reflected upon, and represented.

3.2.7 Communication, coordination, languaging, behaviour and expectations

Maturana and Varela argue that communication is a coordination of behaviour (Maturana and Varela, 1987, page 196). Language arises when there is communication about communication, and 'languaging' takes place when there is a coordination of coordinations of behaviour (Capra, 1996, page 280). Capra illustrates this as follows:

"Suppose that every morning my cat meows and runs to the refrigerator. I follow her, take out some milk, pour it into a bowl, and the cat begins to lap it up. That is communication - a coordination of behaviour through recurrent mutual interactions, or mutual structural coupling. Now suppose that one morning I don't follow the meowing cat because I know that I have run out of milk. If the cat were somehow able to communicate with me something like 'Hey, I've now meowed three times; where is my milk?', that would be language. Her reference to her previous meowing would constitute a communication about a communication and thus, according to Maturana's definition, would qualify as language." (ibid., page 280 - 281)

This conception of communication and language emphasises the importance of ongoing coordination of language games so that communication remains meaningful. We not only communicate so that behaviour and expectations can become coordinated, but we also communicate about communication so that language behaviour can become coordinated.

3.2.8 Conduit metaphor

The common conception of communication is that it involves a transmission of information, and can be understood in terms of Reddy's "conduit metaphor" (Reddy, 1993, page 168). According to this conception, communication is something generated at a certain point, this something is usually referred to as information, the information travels along an communication channel which can be seen metaphorically as a tube or conduit, and the information is received by a receiver. Reddy offers the following framework:

1. Language functions like a conduit, transferring thoughts bodily from one person to another.
2. In writing and speaking, people insert their thoughts or feelings into words.
3. Words accomplish the transfer by containing the thoughts or feelings and conveying them to others.
4. In listening or reading, people extract the thoughts and feelings once again from the words. (Reddy, 1993, page 170)

This influences us to speak of 'information' as being 'contained' in a picture, message or book. Maturana and Varela assert that this metaphor is basically false because it emphasizes the sender and the information rather than the mutual orientation of both parties involved in the communication:

"...there is always ambiguity in a communicative interaction. The phenomenon of communication depends on not what is transmitted, but what happens to the person who receives it. And this is a very different matter from "transmitting information"." (Maturana and Varela, 1987, page 196)

There is evidence that the conduit metaphor is deeply entrenched within the English language and that English speakers' 'communications about communication' are biased toward semantic structures consistent with the conduit metaphor. Reddy gives a host of examples to show how widespread the conduit metaphor is in our talk about communication. Here are a few:

Try to *get* your *thoughts across* better.
None of Mary's *feelings came through to* me.
You still haven't *given me* any *idea* of what you mean.
Whenever you have a good *idea* practice *capturing it in words.*
(Reddy, 1993, page 166 - 167)

Lakoff and Johnson provide this sample:

It's hard to *get* that idea *across* to him.
I *gave* you that idea.
Your reasons *came through* to us.
It's difficult to *put* my idea *into* words.
Try to *pack* more thought *into* fewer words.
You can't simply *stuff* ideas *into* a sentence any old way.
The meaning is right there *in* the words.
Don't *force* your meanings *into* the wrong words.
His words *carry* little meaning.
The introduction *has* a great deal of thought *content.*
Your words seem *hollow.*
The sentence is *without* meaning.
The idea is *buried in* terribly dense paragraphs.
(Lakoff and Johnson, 1980, page 11)

These are examples of the sort of dead metaphor that has become entrenched within the English language. Reddy points out that we do not literally "get thoughts across" nor "receive" feelings. This smacks of telepathy and erroneously suggests that communication somehow transfers thought processes bodily (Reddy, 1993, page 166). In Reddy's words:

"The logic of the framework we are considering - a logic which will henceforth be called the *conduit* metaphor - would lead us to the bizarre assertion that words have "insides" and "outsides". After all, if thoughts can be "inserted", there must be a space "inside" wherein the meaning can reside. But surely the English language, whatever metaphysical meanderings it may have been guilty of thus far, cannot have involved us in this kind of patent nonsense. Well, a moment's reflection should nudge anyone into remembering that "content" is a term used synonymously with "ideas" and "meaning" And that recollection is quite meaning-full (sic) in the present context. Numerous expressions make it clear that English does view words as containing or failing to contain thoughts, depending on the success of the speaker's "insertion" process." (ibid., page 168)

3.2.9 "World re-maker" metaphor

Reddy feels that the conduit model of communication "objectifies meaning in a misleading and dehumanizing fashion". His empirical investigations lead him to believe that at least 70% of our speaking about communication is directly in terms of conduit metaphor (ibid., page 177). It influences us to talk and think about thoughts as if they had some kind of external, intersubjective reality such as lamps and tables. It promotes the idea that 'the meaning is in the message', which is fallacious. The word 'message' gets confused with the word 'signal'. The signal, which is no more than a cluster of sounds or marks, is the thing that travels, not the message. You do not 'get the message' but reconstruct it from the signal, and this is a difficult, creative task. Success in reconstructing messages depends on pre-understandings and

assumptions of the worlds of other people. In the face of a breakdown of meaning, the need for a new metaphor for communication is clearly expressed by Lakoff and Johnson:

"When it really counts, meaning is almost never communicated according to the conduit metaphor, that is where one person transmits a fixed, clear proposition to another by means of expressions in a common language, where both parties have all the relevant knowledge, assumptions, values, etc. When the chips are down, meaning is negotiated: you slowly figure out what you have in common, what it is safe to talk about, how you can communicate unshared experience or create a shared vision." (Lakoff and Johnson, 1980, page 231-232)

Often the forms of life and language games of the communication partner are not completely understood, so that reconstructing a message also requires a reinterpretation or re-making of the world of the communication partner. I will call this the "world re-maker metaphor" (which is very similar to Reddy's "toolmakers paradigm" (Reddy, 1993, page 171)).

A critical difference between the conduit metaphor and the world re-maker metaphor is that the former implies "success without effort" (ibid., page 174) while the latter assumes that communication will always be problematic to some degree because of discrepant contexts, worlds, forms of life or language games. The world re-maker expects to expend a great deal of effort in making sense of his counterpart's world and the environment of the signals he receives. He will devote a great deal of effort to testing his re-makings for accuracy. The conduit metaphor localizes effort in the sender who constructs the message which, having been sent, is assumed to be well received. Because the conduit metaphor assumes the best possible outcome, the participants are not motivated to make sense of each other's worlds, and the shared interpretation system deteriorates as ever more confusing messages start to move back and forth. The world re-maker metaphor prepares communicators for the worst, and communication partners are motivated to work hard at repairing and improving their shared interpretative system.

The conduit metaphor represents a major challenge to the designer because the logic of its framework is spread throughout the syntactic and semantic fabric of our habitual speech acts. Being aware of another metaphor is all very well, but a crucial language game, the English language itself, riddled as it is with the semantic pathologies resulting from the conduit metaphor, is an umbrella language we are forced to use. (ibid., page 176). Mass communication is likely to compound the problem as it becomes synonymous with less meaningful communication:

"We have the mistaken, conduit-metaphor influenced view that the more signals we can create, and the more signals we can preserve, the more ideas we "transfer" and "store". We neglect the crucial human ability to reconstruct thought patterns on the basis of signals and this ability

founders..... the conduit metaphor is leading us down a technological and social blind alley. That blind alley is mass communications systems coupled with mass neglect of the internal, human systems responsible for nine-tenths of the work in communicating. We think we are "capturing ideas in words", and funneling them out to the greatest public in the history of the world. But if there are no ideas "within" this endless flood of words, then all we are doing is replaying the myth of Babel - centering it, this time, around a broadcasting tower." (ibid., page 188)

3.2.10 Final vocabulary

For convenience this categorisation will be dealt with in section 3.3.6.

3.3 THINKING

In section 2.2, the Chaos Report on software project failure highlighted the need for involvement, support, ownership, clear requirements, realistic expectations, and competent, focused project members. This implies that success in the software process depends greatly on an ability to form cohesive project communities. Systems thinking is particularly concerned with this issue. Churchman, in a preliminary statement about systems design, argues that "Design enables us to create systems which will perform tasks better than a single person alone" (Churchman, 1971, page 3). This immediately implies the idea of designing collaborative communities. Churchman is concerned with the manner in which communities form around shared issues, a process which he calls "making polis" (Churchman, 1979, page 74). He is also concerned with the design of inquiry, the activity which produces knowledge. Churchman's work, which focuses on systems inquiry and systems design, is very relevant to the designer who is viewing the organisation as a conversation. In particular, his discussion on the designs of inquiring systems can be interpreted as being designs of conversations to organise knowledge and generate the issues that encourage communities to form and purposeful projects to be designed.

Churchman has also influenced the development of systems practice although he himself did not develop a formal methodology. Methodologies which draw heavily on his influence tend to be classified as representative of soft systems thinking. Flood and Jackson argue that understanding Churchman's work assists greatly in understanding the whole soft systems tradition, including SAST (Strategic Assumption Surfacing and Testing) (Mason and Mitroff, 1981), IP (Interactive Planning) (Ackoff, 1981), SSM (Soft Systems Methodology) (Checkland, 1981) and CSH (Critical Systems Heuristics) (Ulrich, 1983) (Flood and Jackson, 1991, page 120). Jackson gives the following generalisation about the soft systems trend as a whole:

"The emphasis in soft systems thinking is on how to cope with ill-structured problems or messes. Rather than attempting to reduce the complexity of messes so they can be modelled mathematically or cybernetically, soft systems thinkers seek to explore them by working with the different perceptions of them that exist in peoples' minds. Systems are seen as the mental constructs of observers rather than as entities with a real, objective existence in the world. Multiple views of reality are admitted and their implications are examined. Values are included rather than excluded (in theory) from the methodological process. The privileged role of experts is questioned and an attempt made to include problem-owners and other concerned individuals in carrying out the study and finding possible ways forward. The immediate aim is to reach an "accommodation" about action to be taken. This should emerge from a debate involving all those interested in the decision and its implementation. A longer-term objective is to encourage and institutionalise a process of continual learning among the participants of the social system being addressed." (Jackson, 1991, page 296)

Soft systems methodologies can all be seen as conversation designs. Our interest in the systems approach stems from breakdowns in the domain of acting and the design of conversations to cope with these breakdowns. Collaborations often become "messes" or "situations that consist of complex systems of strongly interacting problems" (Ackoff, 1994, page 211). For example, if the shared system of interpretation breaks down, there may no longer be an ability to communicate meaningfully. Fragmented forms of life, incoherent language games, misinterpreted or incomprehensible speech acts, and uncoordinated commitment networks will result in breakdown beyond the scope of a simple solution. There is a potential for this kind of mess to develop in any new project with people from diverse forms of life, speaking different languages, not sure of the issues or purpose nor how they feel about them. A conversation is needed to build or rebuild the foundation of meaning. The systems approach of Churchman provides a good grounding for a designer faced with such a task.

3.3.1 The systems approach, knowledge, image, community, shared purpose

Churchman's inquiring systems can be seen as designs of conversation to generate, organise and synthesize knowledge. "Inquiry is an activity which produces knowledge" (Churchman, 1971, page 8) and learning is fundamental to this process. The task of inquiry is to produce pragmatic knowledge that enables change and freedom to act (ibid., page 10 - 12). The term 'Image' may be used instead of 'knowledge'. Knowledge has an implication of validity and truth whereas image is what is believed to be true and it is this that largely governs behaviour (Boulding, 1956, page 5-6).

Churchman developed five designs of inquiry systems each based on a philosophical tradition. The Leibizian inquirer is based on the rationalistic philosophical tradition, and builds knowledge by constructing a systematic network of facts, or "fact nets". It emphasises logic and the manner in which knowledge 'hangs together' in a coherent relationship. The Lockean inquiring system represents the empiricist philosophical tradition, and attempts to limit the acceptance of facts into the fact net by using observation as a filter to accept only those facts that have some observable validity as agreed by a community. A Lockean inquirer takes "his first steps toward becoming social rather than merely logical, by asking how he could design a "community of minds" (the so-called Lockean community) which agree about their sensory responses" (Churchman, 1971, page 97). Thus the Leibnizian and Lockean inquirers represent two different designs of conversation, one driven by the need for logical coherence and one driven by the need for social agreement about observations.

Kantian inquiring systems emphasize that information is interpreted through presupposed modes of representation. Modes of representation will determine not only what is noticed, but also the interpretation that is made of elementary data. Churchman gives an illustration in which an observer with an appreciation of chess tries to interpret the game of checkers:

"....imagine that the observer of these two players, although unfamiliar with checkers, is very familiar with chess. He sees "directly" what is relevant; the board has become a chessboard, and the pieces are not chess pieces but are all of the same type. Such an observer sifts out a great deal of irrelevancy, e.g. the conversation between the two players, the time between the moves, the behaviour of other people, and so on. The inquiring system that knows chess would receive only those inputs from the movements of the two players that were deemed relevant once the game was taken to be a kind of chess game. The very first move would be interpreted as a bishop move, and the inquirer would start by "representing" the game as a game of twelve bishops on each side. The inquirer would receive very few surprises. It would soon guess by simple deduction that the bishops are constrained to forward, one-place moves; it would see that the rule of "taking" is modified, and so on. .....

We now note that the mode of representation of information seems to strongly influence the success or failure of the inquirer in arriving at a solution." (ibid., page 138-139)

There are, however, many other equally appropriate ways of representing the game of checkers other than as a specialised game of chess. Thus, problem solving involves two aspects. Firstly, there is a search for a pathway that leads from the given to the solution. Secondly, there is an attempt to formulate the given in such a way that this pathway will be easy to find (ibid., page 140). In other words, once a successful mode of representation has been found, the problem becomes much simpler to solve (ibid., page 139). The Kantian inquirer is concerned with a conversation designed to establish the mode of representation that affects the receiving of inputs and discovering its presuppositions (ibid., page 129).

The Hegelian inquirer is concerned with a conversation that establishes objectivity through a dialectical learning process. It gives up the "supremacy of privacy" (ibid., page 155). Strongly held private views are subjected to public scrutiny and synthesised with opposing views. Recognizing that there are many different ways of representing issues, public objectivity is sought through comprehensiveness. This is achieved by sweeping in as many opposing views and observations as possible, and attempting to find the interconnections. Churchman sees objectivity as:

".....a collection of interconnected observations in which each observer can examine how another observer views the world. The objectivity of experience is to be based on some kind of interconnection of observers." (ibid., page 149).

In order to introduce its dialectic nature, it is useful to compare the Hegelian inquirer with the other inquirers:

"... the Leibnizian inquirer ... consists of a stream of sentences ... some of which may be true, others false, others irrelevant. The citizens' problem is to put together several consistent stories and then, as the data flow increases, to converge on one story that seems to hold together in the best manner. The Lockean inquirer displays the "fundamental" data that all experts agree are accurate and relevant, and then builds a consistent story out of these. The Kantian inquirer displays the same story from different points of view, emphasizing thereby that what is put into a story by the internal mode of representation is not given from outside. But the Hegelian inquirer, using the same data, tells two stories, one supporting the most prominant policy on one side, the other supporting the most prominant policy on the other side. The teleological issue is: Which method of telling the story will produce the optimally informed citizen when each is constrained by the same cost and time resources?" (ibid., page 176 - 177)

This highlights storytelling, representation and interpretation. Stories are told according to a world view or "*Weltanschauung*". Whether a piece of information is valid or objective depends on the chosen *weltanschauung* within which it is interpreted, what purpose it serves and its usefulness as a guide to action (ibid., page 170). The Hegelian inquirer constructs a thesis by first acquiring "as broad a sweep of "data" as is possible" (ibid., page 170) and then "showing that there is a way to look at reality so that the data can be interpreted to support the thesis" (ibid., page 172). The dialectic is constructed by developing an antithesis which in effect opposes the thesis becoming its "deadliest enemy" dedicated its destruction (ibid., page 172). The opposition between the thesis and antithesis is observed and a new world view is constructed which makes the nature of the conflict understandable and enables it to be "devoured by the higher-level Weltanschauung" (ibid., page 174). This resolution, often referred to as the synthesis, may in turn become the thesis for a future dialectic. In this way models expand and become more refined. Churchman describes his conception of objective reality as follows:

"....an approach to reality based on the most forceful arguments and counter-arguments at each stage must in the end have eliminated every conceivable ground for doubt. The world will have been examined from every possible point of view - i.e. "objectively" . (ibid., page 175 - 176)

In developing the Singerian inquiring system, Churchman is making explicit the ideal of progress that is implicit in the Hegelian inquirer (ibid., page 178). It also emphasizes co-operation within a community. Singer's idea of progress involves indefinitely approaching an approachable but unattainable ideal (Singer, 1945, page 75):

"... it follows that the self-interest of every member of a progressive community engages him to make whatever sacrifice is necessary to further the community along two distinct but mutually helpful lines of progress: the one, moral progress toward an ideal world of co-operative perfection; the other, scientific progress toward an ideal world of collective omniscience. Neither of these goals is to be conceived as attainable; were either so, it would not be an ideal. But it must not be forgotten that a well-defined ideal is a no less practical guide to conduct than the most attainable of objectives;" (Singer, 1945, page 30).

The Singerian inquirer introduces a conversation designed to unsettle the agreement so carefully synthesised by Hegelian conversations. It does this in the spirit of progress so that, when there is agreement and when data and hypotheses are mutually compatible, it then becomes the time to "sweep in" new information in order to generate dissent. New data or more precise measurements may require an adjustment to the prevailing image or world view. When there is a well-established theory, there should be active encouragement to produce equally plausible counter-theories so that the dialectic can refine and expand the models that are accepted as the status quo. The Singerian inquirer stimulates attack on well-accepted paradigms for the sake of progress (Churchman, 1971, page 199). The Singerian inquirer is teleological, its purpose being to create knowledge for the betterment of all. In this way it introduces an ethical component into conversations.

Churchman sets out the following conditions as being necessary in order for something to be conceived of as a system:

1. The system is *teleological*
2. The system has a *measure of performance*
3. There exists a *client* whose interests (values) are served by the system in such a manner that the higher the measure of performance, the better the interests are served, and more generally, the client is the standard of the measure of performance
4. The system has teleological *components* which coproduce the measure of performance of the system
5. The system has an *environment* which also coproduces the measure of performance
6. There exists a *decision maker* who - via his resources - can produce changes in the measures of performance of the system's components and hence changes in the measure of performance of the system
7. There exists a *designer* who conceptualizes the nature of the system in such a manner that the designer's concepts potentially produce actions in the decision maker, and hence changes in the measures of performance of the system's components, and hence changes in the measure of performance of the system
8. The designer's intention is to change (*implement*) the system so as to maximize the system's value to the client
9. The system is stable with respect to the designer in the sense that there is a built-in *guarantee* that the designer's intention is ultimately realizable

(ibid., page, 43)

Churchman later categorised these nine conditions into "planning categories" made up of three groups of three (Churchman, 1979, page 79 - 80). Table 6 details these groupings.

Group 1	Group 2	Group 3
Client	Decision maker	Planner
Purpose	Components	Implementation
Measure of performance	Environment	Guarantor

Table 6. Churchman's planning categories.

Churchman gives the following explanation of his planning categories:

"The categories are based on the idea that people are the centre of the planner's reality. There are three groups of people: those who should be served (the clients), those who should make the decision (decision makers), and those who should plan (the planners). The categories are thus ethical in kind, but each carries its realistic counterpart, those who are served, do make decisions, do plan. As we'll see, all the remaining categories break down into a "should-is" relationship. The remaining categories are used to fill in the relevant information about each of the three groups. For the client, we need to know what purposes he should have (or has), and how the variety of his/her purposes should be (is) unified (under a measure of performance). For the decision maker, we need to know what he should be (is) able to use as resources (the components of his system), and what he should not (cannot) control which nonetheless matters (the "environment"). Finally, for the planner, we need to know how he should be (is) able to implement his plans, and finally what should be (is) the guarantor that his planning will "succeed" - that is, secure improvement in the human condition." (ibid., page 79)

The planner is concerned with questions of motivation for the system, limits of control and guarantees of success. Group 1 is concerned about the sources of motivation, Group 2 is concerned about sources of control, and Group 3 is concerned about sources of expertize and implementation (Ulrich, 1983, page 258). The planner is also concerned with four "enemies" of the systems approach, namely politics, morality, religion and aesthetics. These enemies resist the systems approach because they wish to be justified on their own terms and avoid being involved in any conversation designed to "sweep" them into a comprehensive, objective, synthesised world view. The designer will try to ensure that the enemies are absorbed in the unfolding process advocated by the systems approach. The enemies are not logical categories but unfold into each other:

"Human history is a tapestry made up of the interplay of the four enemies. Politics is the background of human events, as people have formed themselves into communities and nations. Morality is the deep red hue of revolutions, dissent and heroism. Religion is a pervasive tone, which melds into the background of politics by turning into doctrine and bureaucracy or into morality as the inspiration of religious wars. The history of aesthetics is rarely written, except in histories of art and (occasionally) in biographies, but the true essence of aesthetics is what gives the tapestry meaning." (ibid., page 26)

3.3.2 Background - unfolded and reflected upon, present-to-hand, multiple representations

The systems approach can be seen as a process of unfolding this "background of human events" through various types of conversation and surfacing shared public images from a hidden background of private images. Background is reflected upon and properties, objects and experience emerge into consciousness becoming present-to-hand. Comprehensiveness is sought through multiple representations, and stories about a problem situation are told from as many different points of view as possible. The issues generated are transformed into a shared appreciation of the situation allowing clients, decision makers, planners and other involved stakeholders to become related in communities that are regulated or governed by a common purpose. The systems approach offers the designer insight into the design of learning conversations required to generate shared meaning. The nine conditions of systems design offer the designer a framework to encourage the forming of a viable project community around common issues and shared meaning.

3.3.3 Dialogue, socio-therapy, collective group intelligence, learning, assumptions

The process of unfolding shared meaning is epitomized by Bohm's conception of dialogue. This form of conversation is based on the idea that thought is a collective phenomenon based on how we interact. When thought becomes incoherent, which Bohm sees as the cause of a lot of the world's counter-productiveness, it cannot be improved individually because real intelligence is collective. A mode of conversation is needed that allows access to a larger pool of common meaning found in the collective intelligence of a community. The assumptions, metaphors or images underlying many points of view are surfaced, freely expressed and literally "suspended" before the community so that they can be clearly seen. In this way, dialogue allows people to become observers of their own thinking as well as that of the rest of the community. We forget that problems are created by thought which thought is then called on to solve. By understanding that it is thoughts, as opposed to people, that conflict, individuals can move beyond the traps of their own personal views. Thought can be incoherent if cultural assumptions are invisible or fragmented. Dialogue is a form of collective learning in which a community becomes aware of the incoherence of thought and then moves forward by building shared meaning.

Dialogue is a sustained collective inquiry into the background that we take for granted and it sets up a mode of conversation where people pay attention to the roots of thought and the primacy of the whole. Bohm's introductory remarks on dialogue also help to differentiate it from the conception of communication which is based on the conduit metaphor:

"One meaning of "to communicate" is "to make something common," i.e., to convey information or knowledge in as accurate a way as possible. This meaning is appropriate in a wide range of

contexts. Thus one person may communicate to another a set of directions as to how to carry out a certain operation. Clearly, a great deal of our industry and technology depends on this kind of communication.

Nevertheless, this meaning does not cover all that is signified by communication. For example, consider a dialogue. In such a dialogue, when one person says something, the other person does not in general respond with exactly the same meaning as that seen by the first person. Rather, the meanings are only *similar* and not identical. Thus, when the second person replies, the first person sees a *difference* between what he meant to say and what the other person understood. On considering this difference, he may be able to see something new, which is relevant to both his own views and to those of the other person. And so it can go can go back and forth, with the continual emergence of a new content that is common to both participants. Thus, in a dialogue, each person does not attempt to *make common* certain ideas or items of information that are already known to him. Rather, it may be said that the two people are making something *in common*, i.e., creating something new together." (Bohm, 1996, page 2)

The following quotation relates dialogue to the conception of collaborative projects as "acts of shared creation":

"...if people are to cooperate (i.e., literally to "work together") they have to be able to create something in common, something that takes shape in their mutual discussions and actions, rather than something that is conveyed from one person who acts as an authority to the others, who act as passive instruments of this authority." (ibid. page 3)

Dialogue is a mode of conversation designed to build shared meaning which Bohm sees as being important to the forming of society. The following quotation links to the systems approach which emphasizes the forming of communities governed by common purpose:

"A society is a link of relationships that are set by people in order to work and live together: rules, laws, institutions, and various things. It is done by thinking and agreeing that we are going to have them, and then we do it. And behind that is a culture, which is shared meaning. Even to say that we want to set up a government, people must agree to a common meaning of what kind of a government they want, what's good government, what's right, and so on. Different cultures will produce different functions of government.....

I am saying that society is based on shared meanings, which constitute the culture. If we don't share coherent meaning, we do not make a society." (ibid. page 28)

Dialogue is a conversation designed to look at the roots of thought. Thought fragments the world by dividing things up, thus creating distinctions, categories and words. The things we have selected and separated for our convenience are then given supreme importance and we forget that everything around us has been determined by thought. In Bohm's words, "The point is: thought produces results, but thought says it didn't do it" (ibid., page 10).

The process of dialogue is well illustrated as "socio-therapy", a form of group therapy developed by Patrick de Mare. A group of about thirty to forty people, large enough to represent a microcosm

of society develop trust by talking to one another without apparent purpose. Bohm gives the following example which helps to illustrate the concept:

"Some time ago there was an anthropologist who lived for a long while with a North American tribe. It was a small group of about this size. The hunter-gatherers have typically lived in groups of twenty to forty. Agricultural group units are much larger. Now, from time to time that tribe met like this in a circle. They just talked and talked and talked, apparently to no purpose. They made no decisions. There was no leader. And everybody could participate. There may have been wise men or wise women who were listened to a bit more - the older ones - but everybody could talk. The meeting went on, until it finally seemed to stop for no reason at all and the group dispersed. Yet after that, everybody seemed to know what to do, because they understood each other so well". (ibid., page 16 -17)

A dialogue group "is not wedded to a particular purpose" because it is crucial to create an empty space where there are no limiting assumptions about what is useful (ibid., page 17). This space should create a freedom to express any thought and assumption. These assumptions are not suppressed but suspended for observation so that the whole group acts as a mirror for each person to see how thought processes work. (ibid., page 20 - 21). Bohm describes how collective participation under this setting allow thoughts to flow and link into a whole:

"If we can see what all of our opinions mean, then we are *sharing a common content*, even if we don't agree entirely. It may turn out that opinions are not really very important - they are all assumptions. And if we can see them all, we may then move more creatively in a different direction. We can just simply share the appreciation of the meanings; and out of this whole thing, truth emerges unannounced - not that we have chosen it". (ibid., page 26)

Conviction and persuasion not called for because the emphasis is on the emergence of a "common mind" (ibid., page 27). Bohm's vision of dialogue is that it promotes the forming of a common bond, an impersonal fellowship even in frustrating situations:

"We have to share the consciousness that we actually have. We can't just impose another one. But if people can share the frustration and share their different contradictory assumptions and share their mutual anger and stay with it - if everybody is angry together, and looking at it together - then you have a common consciousness "(ibid., page 33)

Bohm allows that "limited dialogue" where there is a specific purpose or goal in mind may be valuable if the purpose is opened up so that its assumptions can be suspended before the group (ibid., page 42). The problem with dialogue in organisations is that they are normally limited by hierarchy. The associated authority structures limit the ability of people to participate as equals. It is important not to let dialogue lapse into ordinary discussion which emphasizes the primacy of the individual and often degenerates into "advocacy wars" (Senge et al, 1994 page 353). Actions

and purpose are the focus of discussion but merely by-products of dialogue. So, facilitators of organisational dialogue must be awake to this challenge.

3.3.4 Metaphor, conceptual system

Lakoff and Johnson argue that our conceptual systems are structured by a complex but coherent system of metaphors and this is reflected in our language (Lakoff and Johnson, 1980, page 3). Metaphor therefore is primarily a matter of thought and action and only derivatively a matter of language. We act in accordance with this metaphorically structured conceptual system. The conduit metaphor is an example of a metaphor that we live by and shows how a deep metaphor can pervade the domains of thinking, acting and conversing. The significance of this to the practitioner is that metaphorical language in the domain of acting provides a useful window into the metaphorical conceptual system, the domain of thought. By listening to the metaphors that dominate a language game, we can literally 'see' the assumptions underlying thought. Metaphor thus becomes central to meaning, rather than a peripheral aspect of language. We have already discussed the power of language to govern commitment networks in the domain of acting. New metaphors have the power to create new realities (Lakoff and Johnson, 1980, page 145). A new metaphor in the conceptual system will be reflected in the language game and thereby have the power to change the domain of acting. Metaphor acts as an important bridge between the two domains and an important source of organisational change. Lakoff and Johnson describe how metaphor can act as a bridge in mutual understanding and a tool for building meaning:

"When people who are talking don't share the same culture, knowledge, values, and assumptions, understanding can be especially difficult. Such understanding is possible through the negotiation of meaning. To negotiate meaning with someone, you have to become aware of and respect both the differences in your backgrounds and when these difference are important. You need enough diversity of cultural and personal experience to be aware that divergent world views exist and what they may be like. You also need patience, a certain flexibility in world view, and a generous tolerance for mistakes, as well as a talent for finding the right metaphor to communicate the relevant parts of unshared experiences or to highlight shared experiences while deemphasizing others. Metaphorical imagination is a crucial skill in creating rapport and in communicating the nature of unshared experience. This skill consists, in large measure, of the ability to bend your world view and adjust the way you categorize your experience." (ibid., page 231)

Metaphor is put forward in this thesis as a foundation for an additional design of inquiry system. Churchman's five designs of inquiring systems were based on four traditional schools of philosophy: rationalism, empiricism, Kantian criticism (including the Hegelian dialectic), and American pragmatism. The philosophy of meaning provides an additional "school" that emphasizes language and therefore, as will be argued in this section, metaphor. Metaphorical 'truth' and images are as important for establishing shared meaning as epistemological truth and

knowledge. The next subsection contributes metaphoric redescription as a creative method of inquiry appropriate for building shared meaning.

3.3.5 Metaphoric redescription

Rorty argues that truth depends on sentences which are elements of human language. The world is "out there" but truth is not. The world itself does not depend on human mental states, but truth does because it depends on sentences that are elements of human languages. Languages are not found in the world, but are human creations with which we program ourselves and create our beliefs. It is descriptions of the world that may be true or false, but not the world itself. (Rorty, 1989, page 5). Rorty follows a similar argument to Wittgenstein in arguing for the contingency of whole vocabularies:

"When the notion of "description of the world" is moved from the level of criteria-governed sentences within language games to language games as wholes, games which we do not choose between by reference to criteria, the idea that the world decides which descriptions are true can no longer be given clear sense." (ibid., page 5)

Rorty argues that anything can be made to look good or bad by being redescribed, that imagination rather than reason is the central human faculty, and a talent for speaking differently rather than arguing well is the chief instrument of cultural change (ibid., page 7). The following quotation portrays the progression of thought as a battle of vocabularies and also relates vocabulary to action:

"Interesting philosophy is rarely an examination of the pros and cons of a thesis. Usually it is, implicitly or explicitly, a contest between an entrenched vocabulary which has become a nuisance and a half-formed new vocabulary which vaguely promises great things.

The latter "method" of philosophy is the same as the "method" of utopian politics or revolutionary science (as opposed to parliamentary politics, or normal science). The method is to redescribe lots and lots of things in new ways, until you have created a pattern of linguistic behaviour which will tempt the rising generation to adopt it, thereby causing them to look for appropriate new forms of nonlinguistic behaviour." (ibid., page 9)

Rorty, like Wittgenstein, sees language as a tool rather than a medium, and metaphor is central. He describes the development of language as the history of metaphor. Metaphors go in and out of favour depending on their usefulness or appeal. Some find a relatively permanent place in a language game and become part of literal language. Dead metaphors constitute language and become the platform or foil for novel metaphors (ibid., page 16). New forms of life are born from new languages and in time they kill off the old forms of life (ibid., page 19). Rorty emphasises that language and culture are shaped as a result of "a great number of sheer contingencies" and "thousands of small mutations finding niches" (ibid., page 16).

Rorty's ideas on the contingency of vocabularies and language as metaphor link directly to Nietzsche who saw language as a "mobile army of metaphors" (Nietzsche in Kauffman (editor), 1954, page 46). Nietzsche rejected that idea that truth is 'knowable', that reality can be 'represented' by language, and that there is a single context for human lives (Rorty, 1989, page 27). Nietzsche argues that humans relate to the world by creating metaphors:

"What is a word? The image of a nerve stimulus in sounds. But to infer from the nerve stimulus, a cause outside us, that is already the result of a false and unjustified application of the principle of reason... The different languages, set side by side, show that what matters with words is never the truth, never an adequate expression; else there would not be so many languages. The "thing in itself" (for that is what pure truth, without consequences, would be) is quite incomprehensible to the creators of language and not at all worth aiming for. One designates only the relations of things to man, and to express them one calls on the boldest metaphors." (Nietzsche in Kauffman (editor), 1954, page 45 - 46)

Nietzsche's philosophy of perspectivism parallels Rorty's arguments concerning the contingency of language and redescription. The following quotation outlines Nietzsche's position on language and truth, and because it emphasises language use as social convention in a similar manner to Wittgenstein's, it admits metaphor as central to language games:

"What, then, is truth? A mobile army of metaphors, metonyms, and anthopomorphisms - in short, a sum of human relations, which have been enhanced, transposed, and embellished poetically and rhetorically, and which after long use seem firm, canonical, and obligatory to a people: truths are illusions about which one has forgotten what they are; metaphors which are worn out and without sensuous power.....

We still do not know where the urge for truth comes from; for as yet we have heard only of the obligation imposed by society that it should exist: to be truthful means using the customary metaphors - in moral terms: the obligation to lie according to a fixed convention, to lie herd-like in a style obligatory for all..." (ibid., page 47)

Nietzsche's philosophy moves aesthetics and artists, who traditionally are the makers of metaphor, into a central position in the world. The world is justified by the creative activity of artists who create new languages through metaphor (Stern, 1978, page 139). Rorty's interpretation of Nietzsche is that he considered human failure to stem from the acceptance of someone else's description of oneself (Rorty, 1989, page 28). Metaphoric redescription allows an escape from the contingencies of inherited descriptions through recreating oneself in a new language (ibid., page 29). Rorty relates the contingency of language to the pragmatic needs of society:

"We call something "fantasy" rather than "poetry" or "philosophy" when it revolves around metaphors which do not catch on with other people -....Conversely, when some private obsession produces a metaphor which we *can* find a use for, we speak of genius rather than eccentricity or peversity. The difference between genius and fantasy is not the difference between impresses which lock on to something universal, some antecedent reality out there in the world or deep within the self, and those which do not. Rather, it is the difference between idiosyncrasies which

just happen to catch on with other people - happen because of the contingencies of some historical situation, some particular need which a given community happens to have at a given time." (ibid., page 37)

In alignment with Nietzsche's perspectivism, Rorty sees metaphoric redescription as a tool that allows several descriptions of the same event to be juggled without engaging in a battle to determine which vocabulary is 'right'. Metaphors should "rejoice in each other's company" (ibid., page 39). Instead of fighting to replace the vocabularies of others, redescription allows vocabularies to be harmonised by creating metaphors that highlight the relationships between the vocabularies. This difference between 'replacing' and 'connecting' are two different ways of thinking. Finch interprets Wittgenstein as replacing the method of abstract analysis with a method which he describes as "metaphoric connection" (Finch, 1995, page 149). Wittgenstein argued that it is a mistake to dig beneath the surface in order to discover the essence of language. Because language is something that "already lies open to view and becomes surveyable by rearrangement" (Wittgenstein, 1958, paragraph 92), description is more appropriate than explanation as a method for solving philosophical problems (ibid., paragraph 109). Finch's interpretation is that Wittgenstein provides a real alternative to abstract reduction:

"An expression that comes to mind is *metaphoric connection*, a method of unification and generality that is certainly non-reductive and preserves meanings. In fact *it unites in terms of meanings.* A powerful metaphor, one that can grip a whole lifetime or a whole age, may be as universal in range and applicability as any abstraction." (Finch, 1995, page 167)

Metaphoric connection becomes a powerful tool for synthesizing language games and thus important to the design of collaborations. Redescription allows the creation of new metaphors that allow diverse languages to be related and connected. A shared mode of discourse can be created not by destroying vocabularies through analysis, but by creating bridging metaphors. Synthesis comes about by focusing on the shared metaphors that link languages which otherwise seem incomprehensible. Collaborative projects revolve around shared metaphor. If, as I am arguing in this thesis, we make ourselves through language, then it is equally true that we make ourselves through metaphor.

Schon argues that metaphor is a useful way of framing problems. His conception of generative metaphor introduces an explicit insight into the creativity inherent in metaphoric connection and metaphoric redescription. Schon argues that problems are never given, but are constructed by human beings in the stories they tell about the problems. These stories are founded on a "deep metaphor" (Schon, 1993, page 149) that can be uncovered through inquiry and attention. For

example, my argument is that "engineering" is the deep metaphor underlying the software process. An alternative frame is to view software projects as 'conversations'. Schon argues for metaphoric reframing as a way of 'problem setting' which he sees as more critical than 'problem solving'. He argues that debates on social policy are usually more in the nature of dilemmas than problems. The dilemmas are underpinned by conflicting metaphors that cannot be resolved by any appeal to facts, but only by the restructuring and coordination of conflicting frames. Metaphoric restructuring generates new meaning and relationships through the process of "seeing-as" (ibid., page 141):

"the making of generative metaphor involves a developmental process. It has a life cycle. In the earlier stages of the life cycle, one notices or feels that A and B are similar, without being able to say similar with respect to what. Later on, one may come to be able to describe relations of elements present in a restructured perception of both A and B which account for the preanalytic detection of similarity between A and B, that is, one can formulate an *analogy* between A and B. Later still, one may construct a general model for which a redescribed A and a redescribed B can be identified as instances. To read the later model back onto the beginning of the process would be to engage in a kind of historical revisionism" (ibid., page 143)

The metaphors of 'engineering' and 'conversation' are now seen as the competing frames of a dilemma that would be solved through frame restructuring and generative metaphor. Schon describes such a process:

"...frame restructuring and the making of generative metaphor are closely related processes. In both kinds of processes, participants bring to the situation different and conflicting ways of seeing - different and conflicting descriptions. There is an impetus to map the descriptions onto one another, but the descriptions resist mapping. In the context of a particular concrete situation, the participants work at the restructuring of their initial descriptions - regrouping, reordering, and renaming elements and relations; selecting new features and relations from their observations of the situation. As this work proceeds, they represent their experience of the situation through strategies which capture the "next-next-next" of temporal experience of events: and from such representations, of which storytelling is a prime example, they draw the restructured groupings and relations of elements which they are able to embed in a new, coordinated description." (ibid., page 159)

3.3.6 Final vocabulary and "ironical attitude"

Rorty argues that individuals have a "final vocabulary" which is the set of words they employ to justify their actions and beliefs, and with which to tell the story of their lives. It is used to formulate praise, contempt, projects, doubts, hope. Rorty calls it "final" because it marks the limits of an individual's abilities of justification:

"It is "final" in the sense that if doubt is cast on the worth of these words, their user has no noncircular argumentative recourse. Those words are as far as he can go with language; beyond them there is only helpless passivity or a resort to force." (Rorty, 1989, page 73)

An individual who is aware of the contingency of her final vocabularies and the knowledge that anything can be made to look good or bad by being redescribed, is described by Rorty as an "ironist". To be an ironist is to fulfil three conditions:

1. An ironist has continuing and radical doubts about the final vocabulary she currently uses because she has been impressed by other vocabularies which she has encountered.
2. An ironist realizes that the argument phrased in her present vocabulary can neither underwrite or dissolve these doubts.
3. An ironist does not think her vocabulary is any closer to reality than anybody else's.
(Rorty, 1989, page 73)

My argument is that the conversation type of dialogue can be enhanced by an "ironic attitude" because ironists, understanding the contingency and limitations of their final vocabularies, will be more conducive to participating in dialogue than 'non-ironists', who will be inclined to engage in vocabulary battles. The ironist understands that their language game is not a metanarrative grounded in objective reality but a local narrative which is personal and specific to specialised forms of life. "Incredulity toward metanarratives" is Lyotard's simple definition of postmodernism (Lyotard, 1979, page xxiv). Because a collaborative project will potentially comprise a mix of modernist and postmodernist orientations, the designer's challenge will be to develop an ironic attitude whereby people can tolerate redescription of their final vocabularies and not become locked into the analytic defence of metanarratives. Rorty feels that common sense is the opposite of irony, but that common sense holds fast only within the convention of a particular language game. Rorty contrasts the two modes of argument as follows:

"The ironist's preferred form of argument is dialectical in the sense that she takes the unit of persuasion to be vocabulary rather than a proposition. Her method is redescription rather than inference. Ironists specialize in redescribing ranges of objects or events in partially neologistic jargon, in the hope of inciting people to adopt and extend that jargon. The ironist hopes that by using old words in new senses, not to mention introducing brand-new words, people will no longer ask questions phrased in the old words. So the ironist thinks of logic as ancillary to dialectic, whereas the metaphysician thinks of dialectic as a species of rhetoric, which in turn is a shoddy substitute for logic." (Rorty, 1989, page 78)

3.3.7 Humiliation and cruelty

The reason why the ironic attitude is seen as important is that it can help the designer to deal with a collaborative design dilemma. Having made the case for the contingency of language and redescription, Rorty calls attention to a paradoxical effect that this may have:

"But most people do not want to be redescribed. They want to be taken on their own terms - taken seriously just as they are and just as they talk. The ironist tells them that the language they speak is up for grabs by her and her kind. There is something potentially very cruel about that

claim. For the best way to cause people long-lasting pain is to humiliate them by making the things that seemed most important to them look futile, obsolete, and powerless." (ibid., page 89)

This challenges the designer of collaborative projects. A designer attempting to enable progress toward common meaning potentially brings about situations in which people are being explicitly redescribed and therefore humiliated by others. Fostering an ironic attitude and freedom for individuals to redescribe themselves sets up vastly different conversations from situations where individuals are being subjected to humiliating redescriptions by others. Individuals are inclined to retaliate against these cruel redescriptions. Rorty argues that societies are bound together by common vocabularies and common hopes and that typically vocabularies are "parastic" on the hopes (ibid., page 86). Redescription, if it diminishes or fragments a vocabulary, can fragment a community and its hopes. A collaborative project, which aims to build trusting communities which share common meaning, needs careful design in order to thread its way through this paradox.

3.4 CONVERSING

The domains of acting and thinking were grounded in theoretical perspectives on meaning. The domain of conversing lacks similar coherent theoretical foundations and will therefore be built up primarily from research. Whereas the previous sections attempted to build theoretical arguments for various categorizations of collaborative project design, this section will rely ultimately on practitioner's research which will be bolstered by the arguments of specific authors. In order to do this, a practical example of conversation design facilitated by the author will serve to illustrate the vocabulary and categorizations when needed. Additional action research projects will be detailed in Chapter 7.

Schrage, whose work on creative collaboration will be drawn on in this section, argues that "language matters" (Schrage 1995, page 68) and that most management texts overlook this topic. He feels that management texts have a great deal to say on 'communication', but say little about language which is the 'currency' of communication (Schrage 1995, page 71). In other words, the conduit metaphor dominates management literature about communication. This section will argue for an alternative mode of conversation, namely collaboration, which counters the conduit metaphor and is appropriate for use in those situations where the interpretive system not does exist to the extent that meaningful conversation is possible. Collaboration will be defined in narrow terms relative to the needs of the design of collaborative projects. Collaboration as a conversation mode is needed to build on the shared meaning created through dialogue. Collaboration involves extending this shared meaning and using it to design a shared mode of discourse, a meaningful way of talking. As Table 1 shows, the focus of collaboration as a conversation type is to build a basis for viable, understandable and meaningful conversations.

3.4.1 Off-loaded representations, environment, designed reusable mind tools

Dennett argues that the principal difference between humans and other animals is the ability of humans to use "mind tools" (Dennett, 1996, page 100). He argues that there are four kinds of creatures each with a specific way of adapting to their environment. Darwinian creatures survive through chance mutations, by natural selection and survival of the fittest. Skinnerian creatures blindly try different responses until one is selected by reinforcement. The creatures' responses become conditioned. The Popperian creature creates an inner environment that previews candidate acts and selects an action based on certain previewed insights. In other words, based on some kind of mental representation, the Popperian creature increases the chances of its first response being the appropriate response. The Gregorian creature has the ability to import mind tools from the cultural environment; the advantage over Popperian creatures being that the inner

environment can be almost infinitely extended by off-loading it onto the environment in such a manner that it can be later retrieved and imported into the inner environment for future use. Dennett suggests that this off-loading ability is the main reason why human intelligence outstrips that of animals such as whales which have larger brains. The primary source of human intelligence, he suggests, is:

".... our habit of off-loading as much as possible of our cognitive tasks onto the environment itself - extruding our minds (that is, our mental projects and activities) into the surrounding world, where a host of peripheral devices we construct can store, process, and re-represent our meanings, streamlining, enhancing, and protecting the processes of transformation that *are* our thinking. This widespread practice of off-loading releases us from the limitations of our animal brains." (ibid., page 134-135)

This ability allows humans to re-use designs that are part of the cultural environment. Copying designs is "cheap" but inventing new designs is "expensive" (ibid., page 99). When we design a tool, we endow it with intelligence. It requires intelligence to design, fabricate and maintain a tool, and it also confers intelligence on the user of the tool. The better designed the tool, the more information will be embedded in it during its fabrication, and, therefore, the more potential intelligence it confers on its user. These tools are not restricted to physical objects, such as a pair of scissors, but include mind tools. Pre-eminent among the mind tools are words. (ibid., page 99-100) Words allow us to label our environments and thus help to simplify them (ibid., page 135). We also off-load through visual forms of representation such as diagrams, models and maps. The following quotation shows a similarity in Dennett's distinction between know-how and representation and Heidegger's distinction between ready-at-hand and present-at-hand:

"Know-how is a kind of wisdom, a kind of useful information, but it is not represented knowledge. ....some creatures began to refine that part of the environment that was easiest to control, putting marks both inside and outside - off-loading problems onto the world, and just into other parts of their brains. They began making and using representations." (ibid., page 154)

Such representations become objects in their own right, things as Dennett says to be "manipulated, tracked, moved, hoarded, lined up, studied, turned upside down, and otherwise adjusted and exploited." (ibid., page 143)

If we consider the three domains of collaborative projects, we can now distinguish three different states of representation. In the domain of acting where background is taken for granted, there are no mental representations. In the event of breakdown, the domain of thinking unfolds background so that objects and properties become present-to-hand. In the quest for comprehensiveness, multiple mental representations are unfolded. The domain of conversing is intent on establishing a shared interpretative system so that these representations can be used

in viable conversation. The representations are off-loaded from the mind onto the external environment. Here they become public objects with explicit properties and a capability of being manipulated.

3.4.2 Re-representations

Dennett emphasises "re-representation". Representations can be re-represented in other formats for specific purposes. A scale model of a passenger ship can be re-represented as a picture, an accommodation plan, an electrical wiring diagram or a town plan. An organisation, for example, can be represented as an organisational chart or a process dependency diagram or a set of financial statements depending on purpose or point of view. These do not have to be kept in the inner environment of the mind, but can be off-loaded onto the external environment:

"The dramatic improvements in all kinds of investigations, from the foraging strategies of our hunter-gatherer days to the contemporary investigations by our police, poetry critics, and physicists, are due in the main to the explosive growth in our technologies of re-representation.

We keep "pointers" and "indices" in our brains and leave as much of the actual data as we can in the external world, in our address books, libraries, notebooks, computers - and, indeed, in our circle of friends and associates." (ibid., page 144)

Designing for collaboration is concerned with transforming mental representations into representations that exist explicitly in a public environment. This transformation will be enabled by off-loading and re-representing. The public mind tools that are created in this process will endow the collaborative project with greater intelligence and also provide the basis for a shared interpretative system.

3.4.3 Shared space design

Schrage argues that communication is dominated by the "media" metaphor (Schrage 1995, page 15) which emphasises transmission of messages as opposed to understanding messages and sharing thoughts. The media metaphor divorces transmission from the act of creating understanding (ibid., page 22). Schrage suggests that the media treats communication "like an infectious disease" and uses technology to "share an experience" rather than to "create a shared experience" (ibid., page 23).

It is difficult to keep track of what is said in a conversation. Conversations are usually transactional in nature. People take turns to exchange information as opposed to sharing it. Conversations themselves have no memory, even if the participants do. Conversations, therefore, tend to a have a "serial and ephemeral" nature (ibid., page 93). As an alternative, Schrage argues that "shared space" is a necessary element or tool for collaboration. Shared

space changes the dynamics of conversation by adding a dimension which embraces symbolic representation, manipulation, and memory (ibid., page 94). Instead of exchanging information which is primarily a verbal activity, visual modes of conversing are introduced. Representations are off-loaded to become shared objects. Participants will be looking at mutually understood representations and manipulating them while discussing them. The common ground is emphasized rather than a series of individual messages. The point is that something can be built or modified as opposed to merely being discussed.

In order to illustrate shared space, consider a family plus an architect designing a house. The family members are familiar with methods of representing houses via standard building plans. They are capable of sketching rough designs on graph paper, and the architect is capable of understanding these rough sketches and can point out ideas, difficulties or potential problems. With the aid of this shared mode of discourse, they start sketching out rough ideas. The group does not sit around the table and merely discuss ideas about the house, they also employ a visual element. After a while, the group ends up working on a design. They are all focused on it and able to manipulate it by adding lines and rubbing them out. Manipulations can be done concurrently. Once certain boundaries have been established, the top story and the bottom story can be worked on separately but at the same time. Some manipulations may be done away from group discussion. An individual may be given the freedom to design a particular part of the house according to his own requirements. But she is still working in the shared space. There may be much discussion, but ultimately what matters is what ends up on the paper, the shared space.

Schrage argues that shared space allows play, curiosity and serendipity to become part of the conversation (ibid., page 95). One can try things out, push them around in a way which purely verbal conversations do not allow. Shared space heals the rift between spoken and visual languages which should be allowed to work in concert. He feels that "we've divorced representation from human interaction" (ibid., page 95) and that shared space shapes the process of collaboration in much the same way that language shapes the processes of thought (ibid., page 96).

A designer will look for shared space upon which to base collaboration. If it is not naturally available, it will have to be designed. If there is a common means of representing something, such as building plans, then this can be used as shared space especially if it is the type of representation which is conducive to manipulation. There may be many ways of representing something which are well understood, but which are not easily manipulable and therefore poor

design choices for collaboration. For example, a house may be represented in balsa wood as a scale model, but this does not lend itself to easy manipulation. Shared space is more than a representation of something, it must also be able to support a collaborative conversation.

Shared space design may involve more than simply choosing a mode of representation. There are many situations that require collaboration where no obvious shared space exists. The designer will then be facilitating the creation of suitable shared space from scratch. If we take the previous example, but imagine that the house is being built for a blind couple, then the standard shared space, building plans, is not a viable option. If the architect wishes to enable collaboration, she will need to move beyond her familiar mode of representing houses, and become part of a creative process that designs a suitable shared space. In this case, the shared space would need to allow tactile interaction, perhaps Lego or something similar. The architect, as a co-designer of the shared space, would ensure that whatever mode of representation was adopted, it would be such that it could be re-represented as standard building plans.

3.4.4 Private methods

This brings us to a critical issue of extreme importance to collaboration. Designers and collaborators need to make clear distinctions between modes of representation that have the capability of acting as shared space and those which, while they may be perfectly adequate modes of representation, are private methods associated with a specific form of life. The distinction is made clear in the variations of the previous example. In the first instance, building plans are suitable as shared space because all sighted members of the collaboration relate well to the mode of representation and it is also suitably manipulable. The family can literally "build a house" in shared space. The architect's final drawings are merely professional refinements and additions to the initial rough sketches, but the mode of representation remains fundamentally constant throughout the collaboration. In the second instance, building plans were not suitable as shared space for the blind couple, and 'non-standard', tactile shared space was designed. In private, the tactile representation was then translated or re-represented into a standard set of house plans by the architect. In terms of the collaborative project as a whole, these building plans are merely a private method, part of the language game of Western architecture and necessary to a form of life called 'the construction industry'. In the first example, the standard mode of representation can fulfil two functions, namely, acting as shared space in the design *and* construction of a house. In the second instance, the standard mode of representation can fulfil only the latter function as a private method.

Private methods which are used inappropriately as shared space inhibit collaboration. The architect in the second example above may overlook the need to design shared space and attempt to use standard building plans, now an unsuitable private method, as the basis for collaboration. Because the blind couple are unable to manipulate shared space, the conversation will revert from collaboration to communication. The architect will receive information as house requirements, and interpret and record these in an incomprehensible and unshareable private method. The conversation will take the form of a question and answer session with indirect and delayed feedback. The owner of the private method will tend to control the form of the conversation. This is no longer 'an act of shared creation', but an act of solo creation by an architect who receives messages from her clients. This is precisely the dynamic operating in JAD sessions when private I.T. methods, such as data modelling, are attempted to be used as shared space during systems requirement analysis.

3.4.5 Collaboration network

Dealing with private methods represents a challenge to the designer. Forms of life spawn specific vocabularies and methods which become normalized or legitimized through institutionalization. These may support collaboration within a form of life, but may not be able to serve as shared space across a complex web comprising several diverse forms of life. People will tend to use their own familiar vocabularies, but when collaboration spans several forms of life governed by incompatible language games, designing shared space becomes a complex and creative task. Shared space design should not end up as a 'vocabulary battle' invoking humiliation and cruelty. Relegating dominant methods and vocabularies to the status of private methods because they are not suitable for shared space may alter power dynamics. Politics revolves around an ability to control the language game.

Complex collaborations may not be possible with just a single mode of representation. The forms of life may be too diverse and the private methods may be indispensable to each form of life. The collaboration may need to be founded on a number of shared spaces each of which can be related to each other via a network. The collaboration now becomes a system of 'sub-collaborations' each of which has its own shared space that ultimately contributes meaning to the overall project. In our building example, we can image that the blind couple has given the architect a cost limit. The architect might design a collaboration whereby various subcontractors complete a spreadsheet detailing costs of a planned building. The overall project now has two non-standard shared spaces, the tactile model of the house and the contractor's spreadsheet. This collaborative project has now become a synchronised collaboration network comprising the following:

1. A sub-collaboration consisting of an architect and a blind couple. The shared space is designed to allow a tactile means of representing, manipulating and appreciating a proposed house. The collaboration consists of modelling the house. There is a learning mechanism built into this sub-collaboration resulting from the architect reading out costings of various plans from a spreadsheet prepared by subcontractors. Discussion focuses on what is possible as a result of on-going learning about costs. Chosen possibilities are recorded in tactile manipulations of the model.

2. A private re-representation. The architect takes the tactile model of the house and translates it into a standard set of house plans.

3. A sub-collaboration consisting of an architect and a group of subcontractors. The designed shared space is a spreadsheet. The categories within the spreadsheet are determined by referring to the standard set of house plans. The group coordinates and juggles construction options and associated costs. They point at cells on the computer screen, call for new printouts of the spreadsheet, key in new figures in an attempt to make the overall costing fall within budget.

3.4.6 Synchronization mechanisms, conversations and "Conversation choreography"

Winograd and Flores see organisational conversation as a "dance" comprised of speech acts. (Winograd and Flores, 1986, page 64). A collaboration network can be viewed according to this metaphor. A complex collaboration spanning numerous and diverse forms of life requires a complex design. Not only must various shared spaces be designed, but they must also be synchronized so that the collaboration system works as a whole. Sub-collaborations in one part of the network should not produce representations which are incomprehesible to the network as a whole. Re-representations may have to link up certain sub-collaborations. Re-representations may be translations into private methods meaningful to other sub-collaborations, or alternatively re-representations may become the basis for sub-collaborations in their own right. Learning mechanisms may need to be designed so that activity in one sub-collaboration regulates activity in another sub-collaboration.

Extending the dance metaphor, we can call this activity of designing shared spaces and interlinking them through synchronization mechanisms "conversation choreography". This metaphor epitomises the conversation type of collaboration.

3.4.7 Explicated foreground, manipulable representations

In the domain of acting, background was taken for granted and was largely unnoticed. In the thrownness of concernful activity, things were ready-to-hand and not represented in the mind. In the event of breakdown, background is consciously reflected upon and objects and properties become present-to-hand. In coping with breakdown, background is unfolded through dialogue. In the process of building shared meaning, the situation is represented and framed from many perspectives. In the domain of thinking, which focuses on creating shared meaning, there are multiple mental representations of a problem situation. The domain of conversing is intent on translating shared meaning into a shared interpretative system. Once background has been unfolded and represented, certain of these representations are suitable candidates to become shared space. In order to do this, they need to be off-loaded in some manner so that they become explicit. These representations will no longer be the background of activity and conversation. They will become foreground that is explicitly focused upon. In order to be good candidates for shared space, they need to be manipulable. Conversation no longer takes place against a background that is silently taken for granted. Conversation takes place in a foreground of shared space that has been explicitly choreographed and synchronized into a viable collaboration network.

3.4.8 Modes of representation and representation paradox

The design of collaboration is primarily concerned with *modes* of representation. The designer will be challenged by the representation paradox mentioned previously in subsection 3.2.4. In stable situations, experts act in a taken for granted background that is largely unrepresented. Alternatively, a stable form of life may normalize a mode of representation which in turn creates a blindness to other modes of representation which may later become necessary when a previously stable form of life begins to change. In unstable, complex situations or situations where there is a shortage of experts, there is likely to be inadequate modes of representation. In both stable and unstable situations designing new modes of representation is a struggle and an unnatural activity. Making the transition from multiple mental modes of representation suitable for sharing meaning to a few explicit, off-loadable, manipulable *modes* of representation suitable as designs for shared space is the crucial transformation for the designer of collaborative projects. This is crucial to the process of moving from 'shared meaning' to 'cohesive simplicity'. The designer's initial concern is with the design of shared space and not the representations which flow from it. For example, the initial problem in designing a house for the blind couple is not *what* kind of house they want, but *how* to represent any kind of house. Once the mode of representation is designed, the collaborators have a viable and intelligent tool with which to produce the actual representation that they need.

3.4.9 Shared mode of discourse and viable conversations

Designing a mode of representation is a difficult and creative task because, in effect, one is designing a new project specific language. Viable conversation may not be possible without a specifically designed language or vocabulary. Modes of representation set the standards, paradigms and meanings of new language games. The designer is concerned to find objects that can play a meaningful role in a language game. Wittgenstein relates modes of representation, words and language games in the following way:

"Let us imagine samples of colour being preserved in Paris like the standard metre. We define: "Sepia" means the colour of standard sepia which is there kept hermetically sealed. Then it will make no sense to say of this sample either that it is of this colour or that it is not.

We can put it like this: This sample is an instrument of the language used in ascriptions of colour. In this language-game it is not something that is represented, but it is a means of representation.....this gives this object a role in our language-game; it is now a *means* of representation. And to say "If it did not *exist*, it could have no name" is to say as much and as little as: if this thing did not exist, we could not use it in our language game. - What looks as if it *had* to exist, is part of the language. It is a paradigm in our language-game; something with which comparison is made. And this may be an important observation; but it is none the less an observation concerning our language-game - our method of representation." (Wittgenstein, 1958, paragraph 50)

A key intervention, therefore, is the design of an explicitly shared mode of discourse. This involves the design of a new language game interwoven with a new choreographed form of life. In setting up modes of representation, off-loading them so that they are publicly visible standards of comparison, the project will have designed its own interpretative system.

3.4.10 Collaboration and conversation design

This section has categorised and described elements of collaboration as a type of conversation. It has also emphasized the role of the designer as a choreographer of conversations. This metaphor is at the opposite end of the spectrum to the conduit metaphor. The conduit metaphor ignored the space between sender and receiver of information. The conversation choreographer concentrates on how to share this space. The conduit metaphor is concerned with content, whereas conversation choreography is more concerned with finding a viable way to talk. The conduit metaphor's main question is "did you get the message?" whereas conversation choreography asks "how are we going to converse?". Collaboration is focused on establishing a basis for viable, understandable and meaningful conversations.

3.5 THE ACT MOVEMENT PROJECT

Many of the arguments outlined in the previous section stem from practice rather than theory. In this section, a practical application will help to substantiate and illustrate those arguments. This project is also included as a prototypical example of the type of prior practice which inspired further research into collaborative project design. This application highlights collaboration as a conversation type and shows how an unworkable conversation based on the conduit metaphor became transformed into viable collaboration through conversation choreography. In the process of 'turning the conversation around', interventions and insights from all domains are employed, but the primary purpose is to bolster the arguments for the categorizations in Table 1 in the domain of conversing. Documents from the actual project are included in Appendix A.

Fig. 7 outlines the fundamental motivation and task of the ACT movement project. The project took place in a South African life assurance company which was implementing a new life policy administration system called VANTAGE. It was critical that the company's actuaries develop new systems so that they could value policies administered on VANTAGE. At the same time the actuaries wanted enhanced functionality especially in relation to movement data which the old systems, ALIS and EXTRACT, could not provide.

Fig. 8 introduces the principal actors in the project and also describes the prime elements causing an initial breakdown. The analyst felt unable to cope with the project. She could not understand what the VANTAGE experts were saying. In her words, "they talk over my head". Not only was the VANTAGE vocabulary foreign, but none of the analysis tools with which she was familiar could help her to make sense of her situation. This breakdown is *not* an example of the classic IT problem where business users and IT specialists have no common language. The people who she was talking to on the VANTAGE project were systems experts like herself. Specifically, they were systems analysts with an appreciation of life assurance and a competence in software development. The actuaries were also highly computer literate. The analyst would not have been prevented from using her analysis tools with any of these people. But the breakdown on this project was not something that could be solved with analytical tools. This project is simply an example of a conversation that needed some careful re-design.

One of the reasons why this project has been chosen as a detailed example is to illustrate the power of metaphoric redescription and its usefulness to someone who is struggling in a practical situation. I was acting as a mentor to the analyst and also managing a related project. I encouraged the analyst to sort out the problem herself.

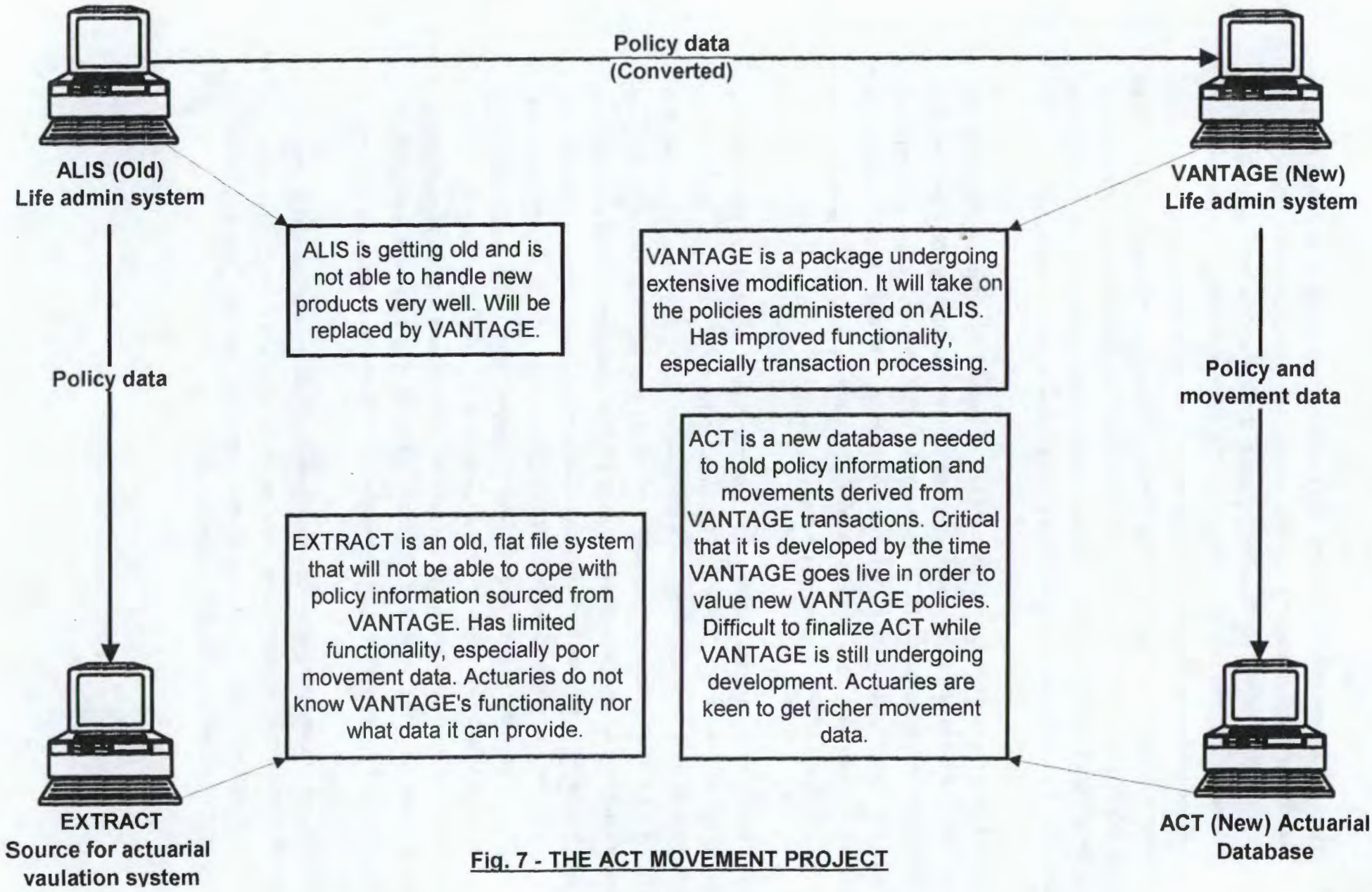


Fig. 7 - THE ACT MOVEMENT PROJECT

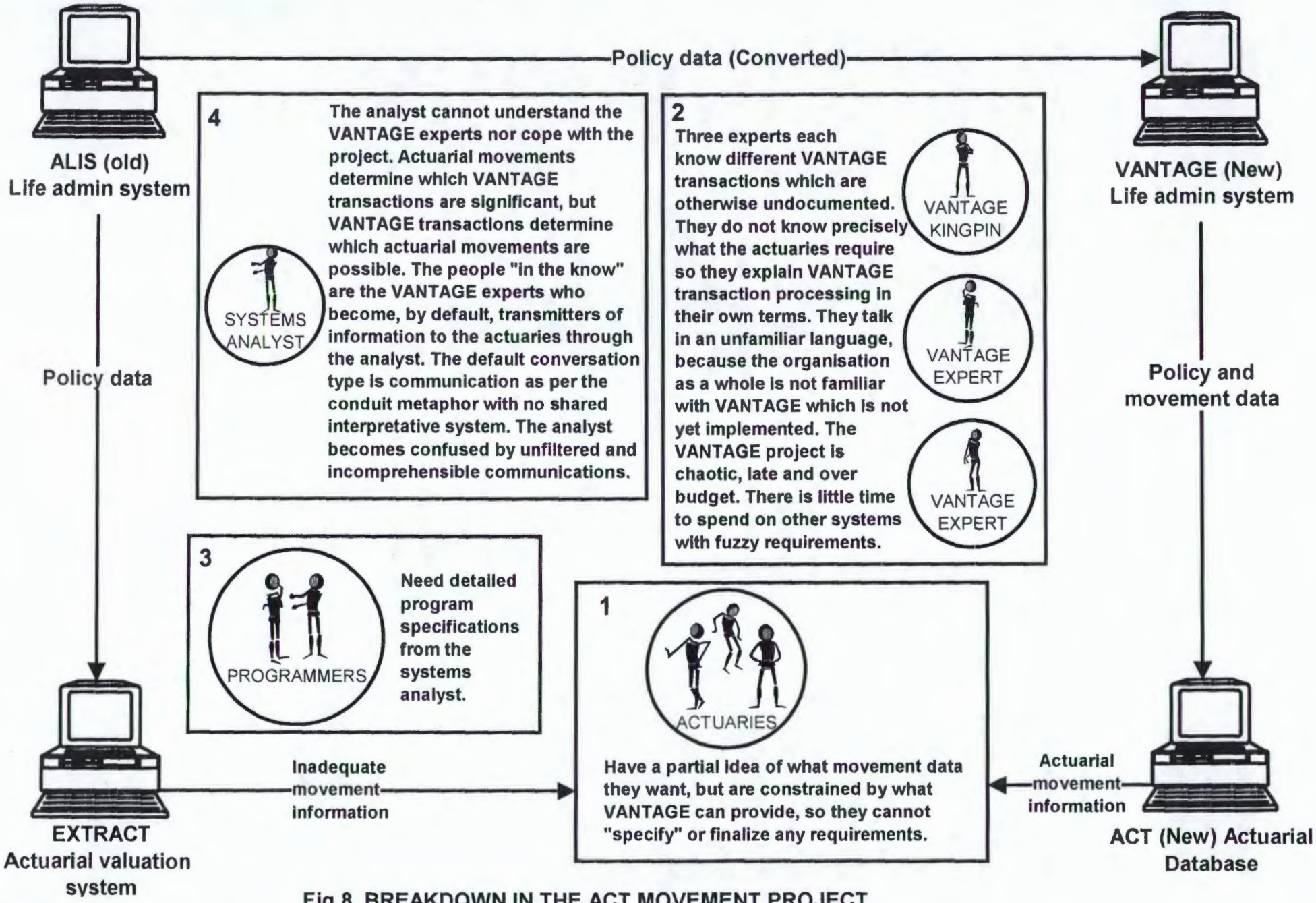


Fig 8. BREAKDOWN IN THE ACT MOVEMENT PROJECT

So the turnaround that occurred on this project was effected by interventions designed mainly by the analyst, but based on facilitation and principles detailed in this thesis. This point is extremely significant because it shows that the categories and distinctions made in this thesis are transferable and useful to other practitioners. Specifically, the analyst was encouraged to reframe her crisis as a 'conversation' problem solvable by design rather than an 'engineering' problem solvable by better analysis.

After reflecting on the poor design of the conversation, the analyst felt that her way to 'get in to the conversation' was to drive it from the actuaries' standpoint. Part of the problem lay in the way the task was set by the actuaries. They were basically saying "Find out about VANTAGE transactions so we can see what information VANTAGE can give us. Once we know this, then we will be able to work with you to specify our movement requirements." Fundamentally, they were setting up a conversation that had to run according to the conduit metaphor. In other words, they were asking her to communicate information about VANTAGE transactions so that they could make some decisions about movements. Under this design, VANTAGE experts would have to initiate the transmissions of information so that actuaries could be the receivers. The analyst would be part of the conduit and would presumably filter the transmissions so that they would be more useful to the actuaries than the "raw" information initially transmitted by the VANTAGE experts. But no such filter could be designed as the actuaries were not providing any design criteria. This in turn meant that the analyst could not construct specific questions to ask the VANTAGE experts. In other words, the system had no learning mechanism. VANTAGE experts could not learn what out of their huge body of knowledge on transactions was the specific knowledge that the actuaries wanted. There were over 300 complex transactions on VANTAGE, but, in the end, less than a quarter turned out to be significant for movement purposes. The only course of action seemed to be to transmit the entire body of transaction information, which is basically what happened in the initial stages of the project. The conduit metaphor assumes a shared interpretation system, but this was absent. The insurance company as a whole did not yet speak "VANTAGE". Few of the initial transmissions were understood by the analyst. Finally, this design wastes resources by reducing the analyst to a helpless and unnecessary link in the conduit. On reflection, the analyst could add no value. Without a filter, a learning mechanism and a shared interpretative system, the VANTAGE experts may just as well have communicated their incomprehensible messages directly to the actuaries.

Once the analyst had metaphorically redescribed her project as a conversation and reflected on the poor conversation design, she had remarkably little difficulty in 'turning it around' to become a collaboration. The first step was to design a filter which set out criteria for identifying movements. Appendix A1 shows this filter. The difficulties of running the project without such a filter were explained to the actuaries, who

then took great care as a group to derive a good set of criteria. This became the 'anchor' for the project and the foundation for the 'movement' language game.

It should be noted that no attempt was made to *define* a 'movement'. Most I.T. methodologies insist on this, but more important is that a word has a meaningful use in a language game (Wittgenstein, 1958, paragraph 29). Lakoff and Johnson point out that categorizations can be based on set theory or prototypes (Lakoff and Johnson, 1980, page 122). Definitions have rigid boundaries, whereas prototypical categories allow various degrees of approximation to a prototypical example. Thus a prototypical chair has four legs, a back, a seat and two armrests. Non-prototypical examples of chairs would be beanbag chairs, hanging chairs, swivel chairs, barber chairs, shooting sticks etc. Appendix A1 shows a cluster of prototypical criteria for movements rather than a finite definition of a movement. In practice, it proved impossible to define a movement and this contributed to the poor initial design of the non-viable conversation. The prototypical criteria associated with the word 'movement' allowed it to become a meaningful word that had practical use in a language game. It might have been possible to eventually find a convoluted *definition* of the word 'movement', something that would satisfy a large number of 'odd' movements, but its abstract and unnatural nature would prevent it from having practical use in the language game. Being able to find the critical word, representing it with its prototypical criteria, using it meaningfully in all project conversations, and setting it up as the foundation of a new project-specific language game, all contributed to 'turning the conversation around'. The project now became 'anchored' to a meaningful actuarial word rather than drifting haphazardly according to the foreign VANTAGE vocabulary.

An IT project based on communication as per the conduit metaphor soon starts to exhibit the symptoms of non-involvement detailed in section 2.2. Such a project is not an act of shared creation. The analyst turned this around by designing two synchronised shared spaces. Now that the word 'movement' had a useful meaning, it became apparent that there were many prototypical movements which could be predicted to exist in VANTAGE. During the group discussion when the word 'movement' was being represented, certain other words were mentioned and repeated. Some of these words seemed to relate to various categories and states of movement. The analyst designed a mode of representing movements using these words as categories of a spreadsheet, which came to be known as 'the movements grid' (see appendix A2). She was also able to include some actual examples of likely movements that she had picked up from the discussion. In other words, a group discussion had unfolded some mental representations which were off-loaded in an explicit representation. The anlayst was not primarily concerned with the accuracy of the movements within the spreadsheet. She was initially concerned with

the mode of representation. But the examples did help to make the categorisations meaningful. A session was organised to present this spreadsheet to the actuaries, the specific purpose being to get them to collaborate in shared space.

The actuaries began to manipulate the spreadsheet. They were comfortable with the overall spreadsheet metaphor, but not happy with some of the categorisations and examples of movements. After a number of sessions, a mode of representation was agreed upon, and the number of movements represented according to this mode of representation had grown substantially. Appendix A2 shows a subsection of the spreadsheet and Appendix A3 shows the full spreadsheet as it stood at the end of the project. This sub-collaboration used the movement grid as shared space for the remainder of the project.

It is important to note that the group itself designed their own mode of representation as a collaboration. The initial attempt by the analyst was never an attempt to be completely accurate. It was an enabler and never a private method to be presented as a solution. It was initially conceived by the analyst as a result of observing the actuaries' language and using the distinctions that they seemed to be making. These distinctions, words or categories were off-loaded into a mode of representation that the analyst felt would be a good reflection of the discussion about the meaning of 'movement'. The analyst was acting as a mirror, reflecting back what she thought she had 'seen' and heard. A correct mode of representation plus correct examples would not have sparked collaboration in shared space. In fact, it was essential to get something wrong so that the actuaries could become involved in manipulating first the mode of representations and then its content. The analyst's task as a designer was to do anything at all that enabled the group to design their own shared space. In this case it happened by providing something that was manipulable, and prompting the group to react in a similar way. It was crucial that the group reacted by off-loading some kind of representation or re-representation rather than by merely communicating a reaction. Allowing the group to redescribe *itself* and design its *own* vocabulary also avoids the humiliation and cruelty that concerned Rorty. The movement grid extended the language game initiated by the representation of the word 'movement'. This sub-collaboration now had a set of project specific words that it could use meaningfully.

The analyst went through a similar process in designing shared space for a sub-collaboration with the VANTAGE experts. Because each expert knew only a subset of the total transactions, there was a series of one-on-one conversations with each VANTAGE experts. The main expert, who we shall call the "kingpin" for convenience, was involved in the design of shared space which subsequently proved acceptable to the other two VANTAGE experts. This proved to be a more difficult conversation to

choreograph. The analyst had already had two confusing sessions with the kingpin. The kingpin was pressurised by the VANTAGE project because so much of that project's knowledge was in his head, and also because it was running late and over budget. The kingpin had other priorities than a system for an external department and could agree to spend only limited time with the analyst. The kingpin was clever and knowledgeable but a poor communicator. This whole scenario intimidated the analyst.

The analyst represented to herself, as best she could, what little information she had gleaned from the two previous discussions. She again concentrated on the mode of representation and looked for the distinctions inherent in the kingpin's explanation. In other words, as an observer of the kingpin's language game concerning VANTAGE transactions, she looked for distinctions that could be used to build a mode of representation. She came up with a mode of representation that highlighted transaction processing on VANTAGE. Timings of transactions were important and as were the links between transactions. Transaction names and codes could be recorded within boxes. Additional explanatory notes were inserted where necessary. Appendix A4 to appendix A7 give examples of the mode of representation that was used as shared space throughout the project when collaborating with VANTAGE experts.

The analyst's strategy was to find a mode of representing to the kingpin the sum total of what she had understood from their two previous encounters. She would prompt the kingpin to correct the mode of representation and its content as a manipulation in shared space. In other words, his reaction should be off-loaded as some kind of representation, not merely a communication. In order to ensure this kind of reaction, the kingpin resolved to be honest about her confusion and highlight the poor design of their previous communications. She could rectify part of the problem now that a filter had been designed; the kingpin could now 'see' what the word 'movement' meant so that only a subset of transactions needed to be discussed. The timing of the third discussion with the kingpin was too early for the actuaries to have finalised their shared space design, namely the movements grid, so this could not yet be used as a learning mechanism. Finally, she made it explicitly clear to the kingpin that she did not understand his language nor his communications, and that they needed to build a shared mode of discourse which would allow them to build something together in shared space.

The kingpin's reaction to this was positive and understanding. The two parties immediately began collaborating by manipulating the representation in shared space in a manner that the analyst had hoped for. The content of the representation was obviously incorrect, but the mode of representation itself was surprisingly appropriate. The analyst and kingpin developed a good working relationship and enjoyed

their on-going collaborations. This was partly due to the fact that the analyst agreed to document *all* VANTAGE transactions as a by-product of the collaboration. There was now some benefit to the VANTAGE project as well as the ACT movement project. Appendices A4 through A7 give samples of the shared space used by the analyst and the VANTAGE experts throughout the project.

Fig. 9 shows the full collaboration network of the ACT movement project. Here are some comments on the synchronisation mechanisms:

1. This "anchored" the project by providing a meaning for the word 'movement'.

A. This represents the meaning of the word 'movement'. See appendix A1.

2. This was made possible by A. The movement grid is a fairly natural extension of a discussion about the meaning of the word 'movement'. See appendices A2 and A3. The movement grid is designed to be a driver for the project. It limits what is talked about and sets the scope of conversation. In other words, the movement grid sets a requirement by specifying which movements must be catered for in the ACT computer system and which VANTAGE transactions are involved. Each categorization in the movement grid implies a set of program rules.

B. The movement grid not only provides shared space for the actuaries' sub-collaboration, but also provides a learning mechanism for sub-collaborations with the VANTAGE experts. As the movement grid starts to fill up with mappings of VANTAGE transactions to actuarial movements, the VANTAGE experts become more discerning about which of their transactions are likely candidates to become actuarial movements.

3. This completes the design of a shared interpretative system for the whole collaborative project. There is now an interlinked language game that spans the whole project. There is an interrelated way of talking meaningfully about both movements and transactions.

C. The project design allows a synchronization between the movement grid and the models of VANTAGE transactions. At this stage, the movement grid has empty cells in which to map VANTAGE transactions, and the transaction models have empty spaces in which to record actuarial movements. The transaction models are more dynamic than the movement grid in that when a movement is mapped onto a

transaction, it indicates the time in the transaction processing that a transaction can become a movement.

4. The shared space proves to be suitable for collaborations with the two other VANTAGE experts, who do not have to undergo the same learning process as the kingpin. When they are brought into the project they have the following advantages:

- they can 'see' what a movement means
- they can 'see' what information the actuaries are after
- they can 'see' examples of movements in the movement grid
- they can 'see' the distinctions that are important, especially the flow and timing of transactions
- there is a way of learning when transactions are likely to become movements
- they can 'see' the logic of the whole project
- conversations are viable from the beginning
- there is a choreographed, shared mode of discourse

5. This is the heart of the synchronization. The actuaries have two manipulable representations. They have an efficient way of learning about VANTAGE transaction processing by looking at the transaction models. The actuarial discussion revolves around whether a movement can be derived from the transactions, and crucially, the appropriate timing. The final decision is recorded on the transaction model. Appendices A4 through A7 show large black circles representing various ACT movements which have been mapped onto certain transactions by the actuaries. At the same time, the actuaries are able to build up the movement grid by categorizing movements in certain ways, based on their new understandings of transactions. They also continue to predict more movements which may reside on VANTAGE, and add them to the movement grid, thus continuing to anchor the project. This is complemented by the VANTAGE experts who, based on feedback that they receive by looking at the movement/transaction mappings and categorizations, are also increasingly able to predict which of their transactions will be suitable candidates for movements. The learning mechanism works in both directions. The co-collaborators are increasingly able to 're-make' each other's worlds.

6. The analyst is able to derive program specifications from the movements grid and the models of VANTAGE transactions. She is able to use a standard IT method, action diagrams, which capture the logic needed for programs to derive actuarial movements from VANTAGE transactions. This is essentially a re-representation into a private method. Appendices A8 and A9 are examples. Interestingly, after a short while, the main programmer was able to work directly off the movement grid and the transaction models, obviating the need to create action diagrams.

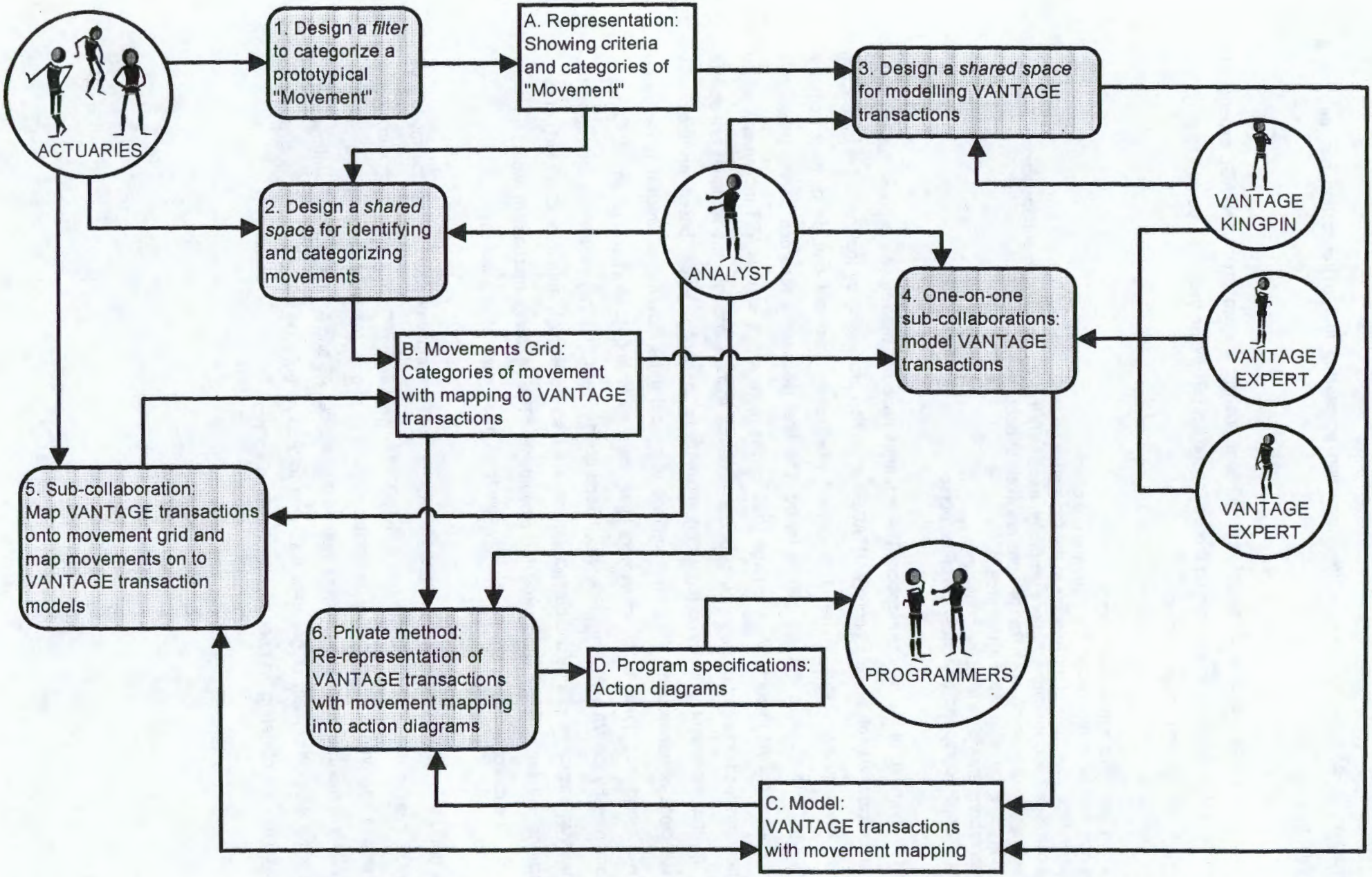


Fig 9. COLLABORATION NETWORK OF ACT MOVEMENT PROJECT

CHAPTER 4. THE CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS

4.1 INTRODUCTION

Fig 1 is a causal influence diagram. It serves a number of purposes. Causal influence diagrams are a useful systems tool for understanding systems dynamics and were used as the basic mechanism for organising the research into collaborative project design. Fig 1 evolved steadily throughout the entire research program. It was built from both action research and literature research. It was used as a tool to design interventions in practical situations. It provided a lens through which to look at various projects and provided a basis for designing the interventions. Its main purpose was to allow a project situation to be seen clearly, in other words, to organise the designer's mind. During each project, the causal influence diagram was amended with the learning experience so that the next project had a richer and more effective lens. Literature research in philosophy, systems thinking, collaboration and the software process was being carried out throughout the research program, and suitable insights were included if and when they seemed appropriate. As mentioned in section 2.10, an important part of the research program was to gain an understanding of the organisational environment of I.T. and especially to understand it as a containing system or the system to be served by software projects. Fig 1 represents a final position on the nature of organisations as a system of dynamic influences. My assumption is that a clear understanding of the essential nature of organisational dynamics provides a sound basis for designing and intervening in collaborative projects.

In order to illustrate the evolution of Fig 1 to its final state, it can be compared with an initial causal influence diagram that was developed early in the research program. It can be seen by comparing Fig 10 and Fig 1 that early versions were focused more directly on the software process itself, whereas Fig 1 is focused on the environment of the software process, and in particular on collaboration. This shift epitomizes the learning that took place.

This chapter corresponds to Step 2 of Schema 1. Step 2 seeks to derive the social process underpinning collaborative projects and then redevelop the social process as a system of dynamic influences. Sections 4.2, 4.3, and 4.4 describe aspects of the social process. Section 4.5 consolidates the description of the social process and represents it as a causal influence diagram in Fig 1. In the process section 4.5 achieves the following:

- it merges or interweaves the vocabulary of the sociology of knowledge (section 4.4) with the vocabulary of Table 1 (i.e. the 12 influencers). In the process it extends the 12 influencers to add a degree of measurability to them (e.g. 'activities' becomes 'variety of activities' which implies that 'variety' can be measured in some way)
- it relates the 12 measurable influencers into a dynamic causal system and represents this system as Fig 1

- it extends the vocabulary of collaborative projects by introducing 14 keywords to exemplify the essence of each influence (the keyword names are finally justified in sections 5.6 and 6.2)

The fold-outs of Schema 1 and Fig 1 should help to orientate the reader and assist in assimilating section 4.5. Section 4.6 identifies a common structure among the influences and their keywords, and represents this common structure in Fig 11. The common structure of the three subsystems is critical to model building in Chapter 5.

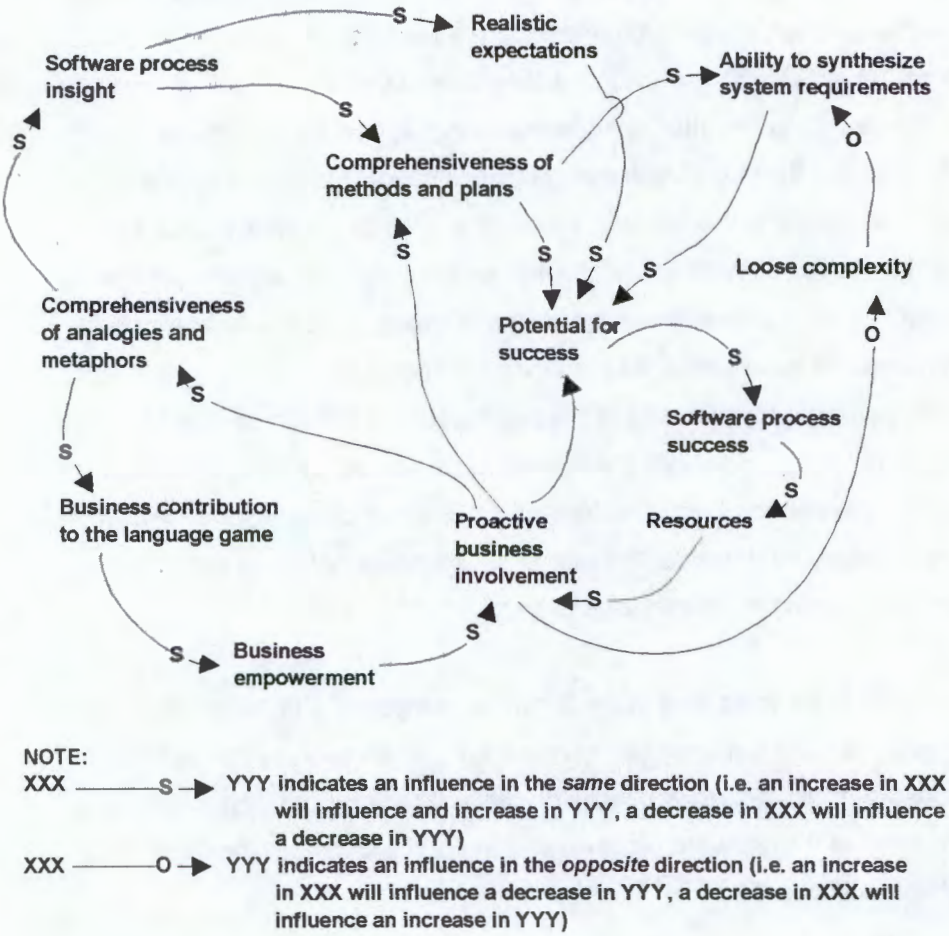


Fig. 10 Initial causal influence diagram

The theoretical justification of Fig 1 will be achieved through descriptive analysis from the standpoint of the sociology of knowledge. The social process inherent in collaborative projects needs to be understood. Table 1 gives a static set of useful distinctions, but it is essential that these can be dynamically related so that the influences can be understood by the practitioner who will be designing interventions to affect these dynamic influences. A descriptive analysis of the process involved in the social construction of reality is applicable to the three domains (acting, thinking and conversing), and therefore provides an appropriate, common overlay. Like the previous chapter, this chapter is concerned with meaning. Table 1 answers the question "what are the essential elements and insights needed by a designer of collaborative projects?". Fig 1 answers the question "how does a group of collaborators socially construct the reality of a collaborative project?".

This chapter draws heavily on the work of Berger and Luckmann for a pure exposition on the sociology of knowledge, and on Boulding whose account of knowledge in society is influenced by general systems thinking.

4.2 THE SOCIOLOGY OF KNOWLEDGE AND THE SOCIAL CONSTRUCTION OF REALITY

In contrast to philosophical inquiry which seeks to establish the ultimate nature of truth, inquiry from the perspective of a sociology of knowledge is interested in what *passes* for knowledge in specific social contexts. Societies differ in what is taken for granted as 'knowledge' within them. A sociology of knowledge is less concerned with establishing ultimate truths than with the general process by which any body of 'knowledge' comes to be established as 'reality' (Berger and Luckmann, 1966, page 15). This position is consistent with the position developed in Chapter 3 through the philosophy of meaning. Berger and Luckmann contend:

"....the sociology of knowledge must concern itself with whatever passes for 'knowledge' in a society, regardless of the ultimate validity or invalidity (by whatever criteria) of such 'knowledge'. And in so far as all human 'knowledge' is developed, transmitted and maintained in social institutions, the sociology of knowledge must seek to understand the processes by which this is done in such a way that a taken-for-granted 'reality' congeals for the man in the street. In other words, we contend that *the sociology of knowledge is concerned with the analysis of the social construction of reality*". (ibid., page15)

Similarly, Boulding argues that truth can best be measured not by correspondence to some external world nor by internal coherence, but by the stability of an image and its ability to survive (Boulding, 1956, page 165 - 168)

Berger and Luckmann argue that theoretical knowledge should not be the central focus of the sociology of knowledge because it is not the central focus of everyday life. Only a very limited group of people in any society are concerned with the theoretical underpinnings of the world, yet everybody lives in a world. What is 'true' does not exhaust what is 'real' for members of a society. Since this is so, Beger and Luckmann contend that common-sense 'knowledge' or everyday 'reality' must be the central focus for the sociology of knowledge. It is this 'knowledge' that constitutes the fabric of meanings without which no society could exist. The analysis of the theoretical part of this reality is, however, a *part* of this concern. Viewed in this light, the scope of the sociology of knowledge moves from the periphery to the centre of sociological theory. (Berger and Luckmann, 1966, page 26 - 27)

"Everyday life presents itself as a reality interpreted by men and subjectively meaningful to them as a coherent world. As sociologists we take this reality as the object of our analyses. Within the frame of reference of sociology as an empirical science it is possible to take this reality as given, to take data as particular phenomena arising within it, without further inquiring about the foundations of this reality which is a philosophical task." (ibid., page 33)

Berger and Luckmann suggest that inquiry into the sociology of knowledge should follow two guiding principles:

"One was given by Durkheim in *The Rules of Sociological Method,* the other by Weber in *Wirtschaft und Gesellschaft.* Durkheim tells us: 'The first and most fundamental rule is: *Consider social facts as things.'* And Weber observes: 'Both for sociology in the present sense, and for history, the object of cognition is the subjective meaning-complex of action'. These two statements are not contradictory. Society does indeed possess objective facticity. And society is indeed built up by activity that expresses subjective meaning. And, incidentally, Durkheim knew the latter, just as Weber knew the former. It is precisely the dual character of society in terms of objective facticity and subjective meaning that makes its 'reality sui generis', to use another key term of Durkheim's." (ibid., page 30)

These two principles can be accepted as the key distinctions that Berger and Luckmann make as *observers* of social reality. In terms of the submarine analogy, an actor 'thrown' in social reality may not make distinctions between "subjective meaning" and "objective facticity", but as observers of social reality, these are distinctions which are made for purposes of sociological description and analysis. In order to describe anything, it is necessary to make distinctions, but these may be more useful and real to the describer than the described . My interpretation is that Berger and Luckmann are *not* conflicting with the philosophical position put forward in Chapter 3. They are *not* saying that there *is* an objective, true world 'out there' which is 'grasped' and represented in the mind as subjective knowledge. On the contrary, they argue that reality is socially constructed, in other words, various societies create various objective facticities through subjectively meaningful action.

The sociology of knowledge inquires into the manner by which subjective meanings become objective facticities, and also the manner in which human activity produces a world of things (ibid., page 30) The world of everyday life is a world that originates in the thoughts and actions of the ordinary members of society, and is taken for granted as reality in the subjectively meaningful conduct of their lives. (ibid., page 33) In the process whereby society constructs a reality in its own image, Berger and Luckmann also emphasize the critical mediating influence of language. They conclude that language cannot be considered a peripheral speciality of little interest to sociological theory, but that it has an essential contribution to make. The sociology of knowledge presupposes a sociology of language (ibid., page 207).

In the context of everyday life, Berger and Luckmann define reality as "a quality appertaining to phenomena that we recognise as being independent of our volition (we cannot 'wish them away')" and knowledge as "the certainty that phenomena are real and that they possess certain characteristics" (ibid., page 13).

4.3 SOCIAL PROCESS AND THE DIALECTIC OF KNOWLEDGE

Society exists as both objective and subjective reality and can be understood in terms of an ongoing dialectical process composed of three moments of externalization, objectivation and internalization (ibid., page 149).

4.3.1 Externalization

Berger and Luckmann argue that externalization is an anthropological necessity grounded in human's biological equipment. Unlike non-human animals which live in closed worlds whose structures are predetermined, man's relationship to his environment is characterised by world-openness. This permits humans to engage in many different activities and to establish themselves in many different environments. Humans are less restricted by their biological constitution to specific environments than other animals and must ongoingly externalize themselves in activity. Humans must themselves provide a stable environment for their conduct, and they themselves must specialize and direct their own drives. This includes creating their own social orders. The necessity for social order stems from human's biological equipment (ibid., page 65-70). In the process of externalization, humans project their own meanings into reality, and thereby construct the world into which they externalize themselves (ibid., page 122).

4.3.2 Objectivation

Objectivation is the process whereby externalized products of human activity attain the character of objectivity (ibid., page 78). The foundation of knowledge in everyday life is the objectivation of subjective processes and meanings by which the intersubjective common-sense world is constructed (ibid., page 34). Objectivation is mediated by language:

"The language used in everyday life continuously provides me with the necessary objectifications and posits an order within which these make sense and within which everyday life has meaning for me. I live in a place that is geographically designated; I employ tools, from can-openers to sports cars, which are designated in the technical vocabulary of my society; I live within a web of human relationships, from my chess club to the United States of America, which are also ordered by means of vocabulary. In this manner language marks the coordinates of my life in society and fills that life with meaningful objects" (ibid., page 35 -36)

4.3.3 Internalization

Individual members of society simultaneously externalize their own being into the social world and internalize it as an objective reality. To be in society is to participate in this dialectic. An individual is born with a predisposition toward sociality, a predisposition to become a member of society. The beginning point is internalization: "the immediate apprehension or interpretation of an objective event as expressing meaning, that is, a manifestation of another's subjective processes which thereby

becomes subjectively meaningful to myself" (ibid., page 149). Socialization mediates internalization of the objectivated structures of the social world into individual consciousness.

4.3.4 Knowledge, Private and Public Images

Knowledge lies at the heart of a fundamental dialectic of society:

"It 'programmes' the channels in which externalization produces the objective world. It objectifies this world through language and the cognitive apparatus based on language, that is, it orders it into objects to be apprehended as reality. It is internalized again *as* objectively valid truth in the course of socialization. Knowledge about society is thus a *realization* in the double sense of the world, in the sense of apprehending the objectivated social reality, and in the sense of ongoingly producing this reality" (ibid., page 84)

Instead of 'knowledge', Boulding prefers to use the term 'Image' :

"Knowledge has an implication of validity, of truth. What I am talking about is what I believe to be true. It is this Image that largely governs my behaviour." (Boulding, 1956, page 5-6)

Image consist of images of values as well images of facts (ibid., page 11). Concerning the 'Public Image', he says:

"The image not only makes society, society continually remakes the image. This hen and egg process is perhaps the most important key to the understanding of the dynamics of society. The basic bond of any society, culture, subculture or organization is a "public image", that is, an image the essential characteristics of which are shared by the individuals participating in the group" (ibid., page 64)

Boulding argues for an organic theory of knowledge in which an organisation grows through an active internal organising principle. A fundamental proposition is that "without a knower, knowledge is an absurdity" (ibid. page 16). His general view of social process "sees the whole movement of society as a process of image-formation under the stimulus of messages transmitted by networks of communication" (ibid., page 98).

Fig. 1 shows three interdependent central influences, 'Cohesiveness of Community', 'Ability to Collaborate' and 'Complexity of Commitment Network'. In order to explain these influences, their make-up and their interdependencies it is necessary to build up an appropriate vocabulary.

4.4 THE VOCABULARY OF THE SOCIOLOGY OF KNOWLEDGE

This section will build a vocabulary for the sociology of knowledge which will outline the social process pertinent to the design of collaborative projects. This vocabulary will be used in section 4.5 to support the causal influences of Fig 1. Each word is selected either because it directly supports the causal influences of Fig 1 or because it contributes to an understanding of meaning which is a central theme of this thesis. (The vast majority of the vocabulary in this section is appropriated from Berger

and Luckmann (1966) with a few supplements from *only* Boulding. Contributions from Boulding will be explicitly referenced. Because *all* the remaining material *without exception* in section 4.4 is attributable to Berger and Luckmann, all statements which are unreferenced are to be read as being implicitly referenced to Berger and Luckmann.)

4.4.1 Intersubjective world

The reality of everyday life involves an intersubjective sharing of the world with others. Existence in everyday life involves continuous interaction and communication with others who share a similar natural attitude concerning the objectifications by which the world is ordered. The perspectives on this common world are not all identical. People have different projects which may even conflict. All the same, people know that they live in a common world and, most importantly, know that there is an ongoing correspondence of meanings. In short, there is shared intersubjective common sense about reality.

4.4.2 Social Interaction

The subjectivity of others is made available through social interaction. Face-to-face interaction is the prototypical mode of social interaction and allows most expressivity, but there are other more remote forms of relating such as correspondence and telecommunication.

4.4.3 Typificatory schemes

Although it is comparatively difficult to impose rigid patterns on face-to-face interaction, interaction is influenced by recurring reciprocal typifications in terms of which others are apprehended and 'dealt with' in face-to-face encounters. These typificatory schemes of interaction become increasingly anonymous the further away they are from the face-to-face situation.

4.4.4 Social structure

Social structure is an essential element of the reality of everyday life and comprises all typificatory schemes and all recurrent patterns of interaction which they establish.

4.4.5 Pragmatic motive

Everyday life is dominated by the pragmatic motive so that, typically, there is little interest in going beyond knowledge that is necessary for pragmatic competence in routine performance. Thus, 'recipe knowledge' occupies a prominent place in the social stock of knowledge.

4.4.6 Relevance structures

The knowledge of everyday life is structured in terms of relevances which may intersect with one another forming a network of relevance structures. As a result, people have things to say to each other and, pragmatically, it is important to have knowledge of the relevance structures of others.

4.4.7 Social stock of knowledge

The social stock of knowledge is socially distributed. It is possessed differently by different types of individuals depending on their expertise, relevance structures and pragmatic need for certain knowledge.

4.4.8 Habitualization of activity

Human activity becomes patterned through habitualization. This enables actions to be performed with economical effort. Choices are narrowed and the individual is freed from the burden of too much decision-making. Habitualization of activity leads to institutionalization.

4.4.9 Institutionalization

Institutionalization occurs whenever there is a reciprocal typification of habitualised actions by types of actors. Institutions are normally collectivities of large numbers of people, but could occur with as few as two individuals. Institutional typification will be characterized by pre-defined situations, assumed reciprocity, patterned conduct habitualized in roles, habitualized communication and an ability to predict the actions of others.

4.4.10 Institutional control

The primary social control is the controlling character of the habitualized patterns inherent in institutions. Additional controlling mechanisms may be needed if institutionalization is not completely successful. Deviance may necessitate the establishment of a system, such as sanctions, to promote compliance and social control.

4.4.11 Institutional historicity

When an institutional world is passed on to a new generation, it is experienced as objective reality with a history and a tradition. The institution is experienced as an objective fact that exists over and beyond the individuals who happen to embody it at a particular moment. At this point it becomes possible to speak of a social world in the sense of reality given to an individual. The institution gains increasing legitimacy with increasing transmission. The nature of the institution needs to be understood in terms of its history. In Boulding's terms, "Part of the image is the history of the image itself" (Boulding, 1956, page 6).

4.4.12 Representation

Institutions are represented or 'made present' in many ways including institutional roles, symbolic objects (both natural and artificial), and linguistic objectifications ranging from simple verbal designations to highly complex symbolizations. All these representations derive their continuing significance and intelligibility from their utilization in human conduct. Where there is a fragmentation of knowledge, there will be problems of binding the various representations together in a cohesive whole that will make sense.

4.4.13 Socialization

Socialization involves the internalization by a new generation of a social reality transmitted by a previous generation. Maximal success in socialization is likely to occur where there is simple division of labour and a minimal distribution of knowledge. Successfully socialized individuals will experience a high degree of symmetry between objective and subjective reality. Berger and Luckmann stress the importance of language in this process:

"Language constitutes both the most important content and the most important instrument of socialization" (Berger and Luckmann, 1966, page 153).

An enormous part of the activity of each society is concerned with the transmission and protection of its public image. A public image almost invariably produces a "transcript"; that is, a record in more or less permanent form which can be handed down from generation to generation (Boulding, 1956, page 64). Language assumes an important role in legitimating the transcript:

"Language provides the fundamental superimposition of logic on the objectivated social world. The edifice of legitimations is built on language and uses *language* as its principal instrumentality" (Berger and Luckmann, 1966, page 82).

4.4.14 Primary socialization

The primary socialization of childhood depends on an emotional identification with significant others, especially parents. This identification mediates an internalization of the parent's world thus bestowing a given identity and a designated social location. In primary socialization there is no distinction between the objectivity of natural phenomena and the objectivity of the social formations.

4.4.15 Secondary socialization

Secondary socialization is a subsequent process that inducts an already socialized individual into new sectors of the objective world of his society. Secondary socialization requires the acquisition of role-specific vocabularies and the internalization of the institutional sub-worlds based on division of labour and concomitant social distribution of knowledge. In order to function in a role and perform its activities, a role specific language must be understood and utilised.

4.4.16 Plausibility structures

Socialization will not be successful if the transcript to be transmitted to subsequent generations is not received as plausible. Complex distribution of knowledge creates various counter-definitions of reality with various social groupings supporting different plausibility structures. When significant others mediate conflicting objective realities, successful socialization is hampered. There may also be situations in which a common reality is mediated, but from considerably different perspectives. As the distribution of knowledge becomes more complex and discrepant worlds become more prevalent, the

significant others mediating socialization will have to concoct a sufficiently cohesive and plausible common world capable of taking on the task of socialization.

4.4.17 Cohesion

The scope of institutionalization depends on the generality of relevance structures. A fragmented institutional order will reflect relevance structures shared by groups within the society but not by the society as a whole. Division of labour leads to a segmentation and dispersion of role specific knowledge. This leads to the problem of providing integrative meanings that will encompass the society and provide an overall context of objective sense for the individual's fragmented knowledge and social experience.

The empirical fact that institutions *do* hang together can be accounted for only in reference to the reflective consciousness of individuals who impose a certain logic upon their experience of the several institutions. There is a need to unite these discrete relevances into a cohesive, meaningful whole.

Boulding emphasises that the aesthetic relationship among the parts of the image may determine its internal cohesiveness by, for example, justifying a highly regarded value system or a way of life. This as well as logical consistency can promote a stability that minimizes the "internal strain" of the image making it more resistant to change (Boulding, 1956, page 13).

4.4.18 Sub-universes of meaning

Socially segregated sub-universes of meaning occur when role specialization is accentuated to the point where role-specific knowledge becomes esoteric against the common stock of knowledge. Sub-universes of meaning promote multiple perspectives often in conflict or competition, which greatly increases the problem of establishing a stable symbolic canopy for the entire society. The increasing number and complexity of sub-universes make them increasingly inaccessible to outsiders and legitimation of these sub-universes for both outsiders and insiders becomes increasingly strenuous.

4.4.19 Reification

Reification of social reality occurs when the products of human activity are apprehended as if they were something other than human products. As soon as the objective social world is established, the possibility of reification is never far away. A reified world is a dehumanized world experienced as a strange facticity over which an individual has no control. Roles may be reified in the same manner as institutions.

4.4.20 Legitimation

Legitimation involves explaining, justifying and giving normative dignity to the values, knowledge and practical imperatives of the institution during socialization. Legitimation is a 'second order'

objectivation of meaning and its function is to make objectively available and subjectively plausible the 'first order' objectivations that have been institutionalized. Legitimation has both a cognitive and a normative element: it tells the individual why he *should* perform certain actions and also why things *are* the way they are.

4.4.21 Language

A special but crucially important case of objectivation is signification, that is, the human production of signs. Language, a system of vocal signs, is the most important sign system of society. The common objectivations of everyday life are maintained primarily by linguistic signification. The common language available for the objectification of experiences is grounded in everyday life and is also used to interpret experiences.

Language, because it can be detached from face-to-face situations, has the capacity to communicate meanings that are not direct expressions of subjectivity 'here and now'. A shared language thus is capable of becoming the objective repository of vast accumulations of meaning and experience, which it can preserve in time and transmit to following generations. Language allows a linguistic community to objectivate shared experiences and is both the instrument and basis of the collective stock of knowledge.

4.4.22 Pragmatic language

Because language originates in and has primary reference to an everyday life dominated by the pragmatic motive, language forces pragmatic patterns of interaction. Prevailing standards for proper speech must be taken into account for specific occasions.

4.4.23 Typified vocabulary

The origins of any institutional order lie in typifications of one's own and others' performances. Forms of action are typified and there will be a vocabulary referring to these forms of action. Division of labour promotes specialized knowledge required to perform particular activities. This fosters a specialised vocabulary which in effect programs the activities. Recipe knowledge and its associated vocabulary are transmitted and sustained via socialization.

4.4.24 Roles

Institutions are embodied in individual experience by means of objectified roles. The institutional order is real only in so far as it is *realized* in performed roles. By playing roles, the individual participates in a social world. By internalising these roles, the same world becomes subjectively real to him. Standards of role performance become known and actors can be held responsible for abiding by those standards.

4.4.25 Organisation

Boulding's conception of an organisation is similar to that of Winograd and Flores who see organisations as conversations:

"An organization might almost be defined as a structure of roles tied together with lines of communication." (Boulding, 1956, page 27).

Boulding likens an organization to an open system in the sense that it maintains its structure in the midst of a throughput of material:

"The social organization maintains its role structure amid a flow of constantly changing individual persons occupying these roles. Men are continually hired, fired, promoted and demoted. They join and resign. They are born and they die. The organizations potentially, at least, go on forever. Organizations like organisms exhibit division of labour, specialization of role, and a hierarchical structure of communication and authority." (ibid., page 27)

The image of the organisation is a property of individual persons, *not* of the organization (ibid., page 28)

4.4.26 Conversation and synchronization

The most important vehicle for reality-maintenance is conversation. (172) In Boulding's words:

"The study of man is the study of talk. Human society is an edifice spun out of the tenuous webs of conversation" (ibid., page 45)

It is important to stress, however, that the greater part of reality-maintenance is implicit, not explicit. Most conversation does define the nature of the world. Conversation takes place against the background of a world that is silently taken for granted. In common daily intercourse we all behave as if we possess roughly the same image of the world (ibid., page 14). Casual conversation is possible when it refers to the routines of a taken-for-granted world.

Conversation enables the synchronization of individual subjectivities. In the face-to-face situation language possesses an inherent quality of reciprocity that distinguishes it from any other sign system. The ongoing production of vocal signs in conversation can be sensitively synchronized with the ongoing subjective intentions of the conversants. Conversation partners speak as they think, and both hear what each has to say at virtually the same instant, which makes possible a continuous, synchronized, reciprocal access to their two subjectivities. Berger and Luckmann elaborate:

"I objectivate my own being by means of language, my own being becomes massively and continuously available to myself at the same time that it is available to him, and I can spontaneously respond to it without the 'interruption' of deliberate reflection. It can therefore be said that my language makes 'more real' my subjectivity not only to my conversation partner but also to myself. This capacity of language to crystallize and stabilize for me my own subjectivity is retained (albeit with modifications) as language is detached from the face-to-face situation. This very important

characteristic is well caught in the saying that men must talk about themselves until they know themselves" (Berger and Luckmann, 1966, page 53).

Because of its capacity to transcend the 'here and now', language bridges different zones within the reality of everyday life and integrates them into a meaningful whole. The transcendences have spatial, temporal and social dimensions. Through language it is possible to transcend the gap between individuals' manipulatory zones. Synchronization becomes possible without face-to-face interaction. Beger and Luckmann emphasize the role of conversation and common language in the creation of social reality:

"This reality-generating potency of conversation is already given in the fact of linguistic objectification. We have seen how language objectifies the world, transforming the *panta rhei* of experience into a cohesive order. In the establishment of this order language *realizes* a world, in the double sense of apprehending it and producing it. Conversation is the actualization of this realizing efficacy of language in the face-to-face situations of individual existence. In conversation the objectifications of language become objects of individual consciousness. Thus the fundamental reality-maintaining fact is the continuing use of the same language to objectify unfolding biographical experience. In the widest sense, all who employ this same language are reality-maintaining others. The significance of this can be further differentiated in terms of what is meant by a 'common language' - from the group-idiosyncratic language of primary groups to regional or class dialects to the national community that defines itself in terms of language." (ibid., page 173)

4.4.27 Universe of discourse

It is discourse or conversation which makes the human image public in a way that the image of no lower animal can possibly be. The term "universe of discourse" has been used to describe the growth and development of common images in conversation and linguistic intercourse (Boulding, 1956, page 15).

4.4.28 Subuniverses of discourse

A public image is a product of a universe of discourse, that is, a process of sharing messages and experiences. Because there are many cultures and subcultures, there is no single public image (ibid., page 132). Subcultures, dominated by differing specializations develop not only an image of their own but also a language of their own. Within the universe of discourse there are, therefore, many specialized subuniverses of discourse. Society is comprised of a series of specialized images on an overlapping continuum (ibid., page 136).

4.5 THE CAUSAL INFLUENCES OF COLLABORATIVE PROJECTS

This section will support the causal influences of collaborative projects represented by Fig 1 using the vocabulary built up in the previous section. In Rorty's terms, this thesis can be viewed as a synthesis or weaving together of a final vocabulary from a variety of complementary vocabularies. The vocabulary of meaning will be linked to the vocabulary of the sociology of knowledge thus providing a sociological bridge or overlay to the positions outlined in Chapter 3 and represented by Table 1. Each

influence will be summarized by a single keyword which represents the essence of the influence. It is important to remember that Fig 1 summarises *all* theoretical readings and action research. The full argument supporting Fig 1 cannot be justified *solely* by the sociology of knowledge. Theory and practice detailed in later chapters also contributed to the final causal influences. This section seeks to add weight to the arguments forming Fig 1 from the perspective of the sociology of knowledge, but the final justification for the choice of the names of the keywords will be accomplished in section 5.6 (as intra-subsystem movements) and section 6.2 (as inter-subsystem movements) once *all* the theoretical arguments have been put forward.

1. *Cohesiveness of community* influences *ability to collaborate* in the *same* direction:
The institutionalization process which produces a cohesive community will encourage a number of characteristics conducive to effective collaboration. A community with a shared public image and shared purpose, with habitualized and reciprocal activity, with a coherent network of relevance structures and subuniverses of meaning is predisposed to collaborate. A cohesive community which performs on its commitments inspires mutual confidence and trust which enhances the ability to collaborate.
Keyword: Trusting

2. *Ability to collaborate* influences *complexity of commitment network* in the *opposite* direction:
A community which can collaborate effectively will be characterized by an ability to converse viably in a shared universe of discourse, or alternatively, synchronized subuniverses of discourse. This implies an ability to explicate clear, meaningful and synchronised commitments. This improves the viability of the commitment network by reducing its apparent complexity.
Keyword: Explicating

3. *Complexity of commitment network* influences *cohesiveness of community* in the *opposite* direction:
When the commitment network is complex, role performance becomes generally more difficult. This may be so not only because the commitments themselves are more difficult to perform, but also because confusion may give rise to misunderstanding about performance standards and expectations in general. The inability of a commitment network to perform expected roles undermines the institutional cohesion and promotes a loss of confidence and trust in the community.
Keyword: Performing

4. *Complexity of commitment network* influences *variety of activities* in the *same* direction:
The variety of activities that each individual performs is influenced via internalization of the explicated commitments which form the network of which the individual is a committed member. Commitment implies an expectation of performance levels to be achieved. Successful socialization of the individual will depend on the commitments being experienced as coherent, plausible, legitimate and meaningful.

It follows that a complex commitment network expands the variety of activities both in number, scope and complexity.
Keyword: Socializing

5. *Variety of activities* influences *variety of forms of life* in the *same* direction:
It is through regularized activity that individuals make a world in common. As the variety of activities increases, externalization and social interaction in the intersubjective world brings forth increased objectivations based on action. The typificatory schemes, pragmatic language, habitualization of activity and development of roles will produce various interdependent forms of life differing in number and complexity. In this way, it is possible to think of a collaborative project as a system comprising a variety of interdependent forms of life. The variety of activities will influence the variety of the forms of life in number and complexity.
Keyword: Worldmaking

6. *Variety of forms of life* influences *complexity of commitment network* in the *same* direction:
An appreciation of a common form of life encourages the forming of institutions through institutionalization. An institution manifests itself in actual experience through the performance of roles. As the variety of forms of life increase the variety of expectations of performance associated with an increasing variety of roles will promote an increasingly complex network of mutual commitments. As the number of commitments increases, it becomes more complex to relate them into a cohesive whole.
Keyword: Forming

7. *Variety of forms of life* influences *variety of language games* in the *same* direction:
A form of life will tend to create a typified vocabulary designed to support its pragmatic motives. The increasing variety of forms of life implies increasing role-specific vocabularies. This happens for pragmatic, practical reasons as specialized motives spawn helpful specialized vocabularies and associated methods. Concomitant increases of sub-universes of meaning and sub-universes of discourse imply an increase in the variety of language games.
Keyword: Designing

8. *Variety of language games* influences *complexity of commitment network* in the *same* direction:
It will be apparent from the descriptions and the vocabulary developed so far that language is the dominating influence in all aspects contributing to the sociology of knowledge. It is fundamental that language, through its ability to objectify, mediates all phases of the social construction of reality. Language governs what can be expressed and therefore what commitments it is possible to make. It is thus a fundamental form of institutional control. As the complexity of language games increases, the complexity of the whole institutional order increases accordingly. It is possible for a word to have different meanings in different sub-universes of discourse. If there is a variety of relatively exclusive

sub-universes of discourse supporting relatively uncohesive universes of meaning, it becomes increasingly difficult to have comprehensible conversations. The commitment network becomes less viable as a result of mounting confusion stemming from the dwindling ability to explicate a coherent network of viable commitments.
Keyword: Governing

9. *Cohesiveness of community* influences *clarity of private images* in the *same* direction:
The private images of each individual are influenced via internalization of the objectified reality of the socializing community. A coherent plausibility structure, a justifiable legitimation story, integrated or minimal sub-universes of meaning, institutional historicity and a generally cohesive social order will influence successful socialization of the individual promoting symmetry between the objective and subjective reality.
Keyword: Socializing

10. *Clarity of private images* influences *clarity of shared public images* in the *same* direction:
Individuals make a common world through externalization of their private images. This worldmaking movement from a subjective world to an intersubjective world is influenced by the form of social interaction. The ability of a private image to have a public existence will depend on the clarifying ability of the language employed to express and objectify it. To be incorporated into the public world, a private image will have to intersect with the relevance structures of other individuals in some pragmatic way. This will be facilitated by clearly expressed private images which enhance the social stock of knowledge.
Keyword: Worldmaking

11. *Clarity of shared public images* influences *cohesiveness of community* in the *same* direction:
Because image governs behaviour (Boulding, 1956, page 6), a clear and commonly shared public image will encourage shared public behaviour and thus the habitualization of activity necessary for institutionalization. The typificatory schemes comprising the social structure will be strengthened by a clearly shared public image. All this encourages the forming of institutions and communities exhibiting natural cohesion.
Keyword: Forming

12. *Clarity of shared public images* influences *clarity of shared purpose* in the *same* direction:
Coherence of relevance structures is influenced by the pragmatic motive. When shared public images cohere because of a coincidence of pragmatic projects in the intersubjective world, awareness of a clearly shared common purpose grows. However, relevance structures seldom intersect perfectly, socially segregated sub-universes of meaning are normal, and pragmatic motives are seldom completely aligned. A sufficiently cohesive common purpose may have to be designed to unite the

disparate pragmatic motives. The degree to which such a relatively shared common purpose can be designed will be influenced by the clarity and degree with which the public image is shared.
Keyword: Designing

13. *Clarity of shared purpose* influences *cohesiveness of community* in the *same* direction:
A clearly shared common purpose will influence behaviour according to its pragmatic motives. The whole institutionalizing process will be governed by the controlling directives of common purpose. Primary institutional control, formed naturally through habitualization of activity stemming from shared public image, will be supplemented by deliberate institutional control derived from shared purpose. The degree to which purpose can influence the cohesiveness of community will be influenced by the clarity of purpose and the degree to which it is shared.
Keyword: Governing

14. *Ability to collaborate* influences *viability of conversations* in the same direction:
The ability to collaborate depends on the degree to which the subuniverse of discourse supports viable collaborative conversation. An ability for an individual to participate in conversation implies a familiarity with the mode of discourse which supports the collaborative project. Becoming a viable member of a project may involve being socialized into an unfamiliar mode of discourse. Viable conversation implies a suitable pragmatic language supporting pragmatic motives. An ability to collaborate facilitates the socialization of new generations thus enhancing the viability of conversations.
Keyword: Socializing

15. *Viability of conversations* influences *ability to design a shared mode of discourse* in the *same* direction:
Collaboration requires a shared mode of discourse. For example, in the construction of a house, the architectural building plan orientates all members of the project through design, construction, determination of rates and so on. This is a standard, shared mode of discourse, but in other projects the mode of discourse may need to be designed from scratch as part of the project. The ability to design a shared mode of discourse implies at least some minimum ability to converse. When subuniverses of discourse are highly segregated, individuals will struggle to make a world where meaningful conversation is possible.
Keyword: Worldmaking

16. *Ability to design a shared mode of discourse* influences *ability to collaborate* in the *same* direction:
A shared mode of discourse encourages shared public behaviour and the habitualization of activity necessary for institutionalization. A shared subuniverse of discourse and mutually understood representations encourage the typificatory schemes which allow a collaborative community to form.

With increasing cohesion and suitability of the shared mode of discourse comes increasing ability for a community to collaborate effectively.
Keyword: Forming

17. *Ability to design a shared mode of discourse* influences *effectiveness of synchronization mechanisms* in the *same* direction:
With the advent of a shared mode of discourse comes increasing opportunities for the synchronization of activities. Representations allows face-to-face social interaction to be transcended. Building plans, for example, obviate the need for a great deal of face-to-face conversation because of their ability to orientate all the parties during constructing of a house. The building plan is not only a synchronization mechanism in itself, but also provides a shared background for the design of other effective synchronization mechanisms. The degree to which the mode of discourse is shared among the collaborating community and the degree to which it suits the pragmatic motives of the project, influences the degree to which effective synchronization mechanisms can be designed.
Keyword: Designing

18. *Effectiveness of synchronization mechanisms* influence *ability to collaborate* in the *same* direction:
Mechanisms designed specifically to increase synchronization are essentially forms of institutional control, and will add deliberate governance to supplement the natural control which formed in the process of institutionalization. Effective synchronization mechanisms increase the ability to collaborate.
Keyword: Governing

19. *Variety of forms of life* influences *clarity of shared public images* in the *opposite* direction:
A form of life normalizes shared public images. As the variety of forms of life increase in number and complexity, so too do associated sub-universes of meaning. There will be increased variety of relevance structures and plausibilty structures, in short, increased complexity. This fragmentation will make it increasingly difficult for clearly shared public images to emerge.
Keyword: Normalizing

20. Variety of forms of life influences *ability to design a shared mode of discourse* in the *opposite* direction:
A form of life legitimises specific modes of discourse. As the variety of forms of life increase in number and complexity, so too do associated sub-universes of discourse. The potential for highly segregated, incompatible subuniverses of discourse increases, thus complicating the ability to design a shared mode of discourse.
Keyword: Legitimizing

21. *Clarity of shared purpose* influences *variety of language games* in the *opposite* direction:
As purpose becomes more clearly shared, the community can focus on a narrower range of activities, specifically those activities which clearly support the pragmatic motive. Language is woven into activity, so a community focused on a clear purpose requires less variety in its language games. In practice, consciously limiting language games so that they support a clearly shared purpose is a powerful mechanism of organisational change. An organisation changes when it talks a new language, and becomes effective when it talks a simple language aligned to a clear purpose.
Keyword: Transforming

22. *Clarity of shared public images* influences *ability to design a shared mode of discourse* in the *same* direction:
The ability to design a shared mode of discourse is contingent on the ability of shared public images to be represented. Clear and shared public images are more able to be represented and off-loaded than fragmented, unclear public images. If a public image is able to be represented, the ability to design a shared mode of discourse is greatly enhanced.
Keyword: Representing

23. *Viability of conversations* influences *clarity of private images* in the *same* direction:
The viability of conversation influences the ability for an individual to express herself and to be understood. We know ourselves and others by talking. Viable conversation allows private images to be expressed and therefore to be externalized in the intersubjective world.
Keyword: Expressing

24. *Effectiveness of synchronisation mechanisms* influences *variety of activities* in the *opposite* direction:
Habitualized, typified and reciprocal activity encourages natural coordination within an institution. Sychronization mechanisms are designed to enhance institutional coordination. If the design of these synchronization mechanisms is effective, it will allow a co-ordination of activity in such a manner that unnecessary actions relative to pragmatic purpose are avoided, thus reducing the variety of activities.
Keyword: Coordination

4.6 THE COMMON STRUCTURE OF THE SOCIAL PROCESS SHARED BY SUBSYSTEMS

The previous section described the influences on collaborative projects from the perspective of a sociology of knowledge. The next chapter will use this description to further the development of the model for collaborative project design by overlaying an appreciative systems perspective. Before doing this, we will first note a pattern which underlies the influences depicted in Fig 1.

It can be seen that each of the three subsystems has a similarity about them. Fig 11 shows a common structure and common keywords shared by the subsystems. Numbered from 1 to 4 are four clusters of influencers drawn from Fig 1. Marked alphabetically from A to E are five common modes of influencing drawn from the vocabulary of the sociology of knowledge. The pattern is a summary of social process.

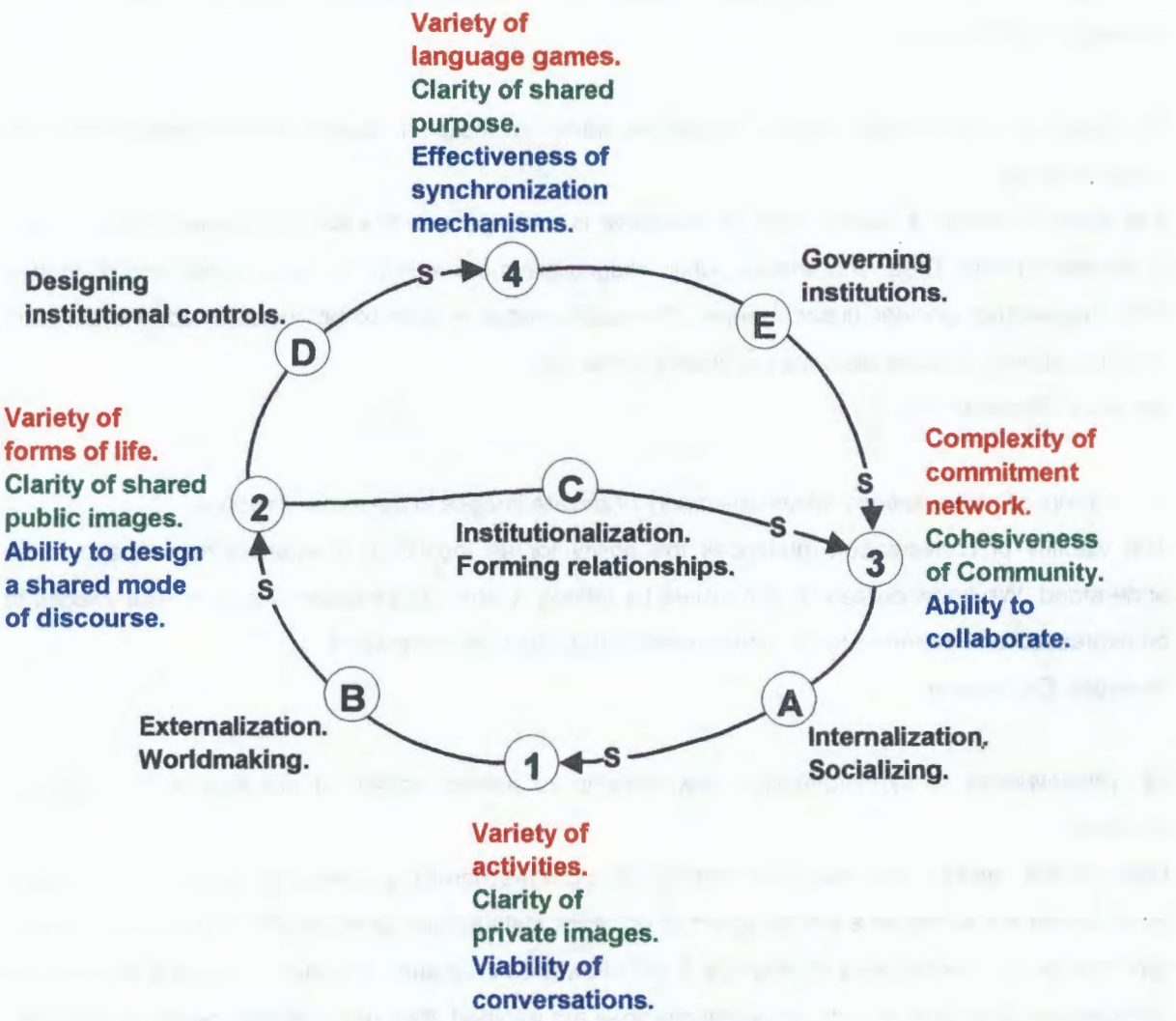


Fig 11. Common structure of subsystems of causal influences

A. An individual internalizes the objectified reality of a community of collaborators who make commitments to each other thus predisposing the individual to certain thoughts, words and deeds. Keyword: Socializing.

1. Individuals subjectively imagine, converse and act. They make choices about what they think, say and do.

B. In their imagining, conversing and acting, individuals externalize themselves through social interaction. Individuals try to make common sense of the world.
Keyword: Worldmaking.

2. A relatively shared outlook on an intersubjective world is objectified through social interaction. Images become publicly shared, an agreed mode of conversing is adopted, and common forms of life evolve.

C. A shared, intersubjective and objectified world encourages a natural process of institutionalization.
Keyword: Forming.

3. Individuals form relationships with each other as members of communities characterised by mutual trust, an ability to collaborate, and an ability to perform on commitments.

D. A shared objectified world in which individuals share pragmatic motives encourages the design of mechanisms for institutional control over and above those which happen naturally as a result of normal institutionalization. The degree to which this is necessary is the degree to which the objectified world is consistently shared.
Keyword: Designing.

4. Conscious agreements are reached on shared purpose, synchronization mechanisms and common vocabulary and methods.

E. Designed institutional controls and effective governance strengthen and formalize the relationships, commitments and trust within the communities.
Keyword: Governing.

This chapter has developed a systemic model of social process which highlights language and conversation as its principal mediators. At this point it is important to remember the reasons why the model is being built and why certain factors or descriptors have been chosen in this chapter or will be chosen in the next two chapters which will extend and complete the model building process. From a systems perspective this thesis is most closely allied to the soft systems tradition which, as has already been stated on page 56, views systems as mental constructs rather than real entities in the world. The model building process is thus attempting to choose, justify and relate various elements, factors or influences into a useful systemic mental construct for a practitioner hold in mind when intervening in projects which require collaboration.

CHAPTER 5. APPRECIATIVE SYSTEMS FRAMEWORK

5.1 INTRODUCTION

Chapter 4 described the causal influences inherent in collaborative projects by using a causal influence diagram, Fig 1, to expound a social process from the perspective of the sociology of knowledge. Influence diagrams by their nature depict dynamic systems and it was an assumption in Chapter 4 that the social construction of reality *is* a dynamic system that *can* be well depicted by an influence diagram. In this manner, a systems tool was used to describe a social process. This research was committed to systems thinking from the outset, and was committed to discovering the usefulness of the systems approach in solving the research problem which was framed initially as the software crisis. Causal influence diagrams were used as a systems tool to organise practical research. In spite of this, however, it is necessary to make a stronger case for the systemic nature of the social process which is being focused on. This chapter will make explicit the systemic nature of the collaborative projects. It will draw heavily on the work of Geoffrey Vickers from which it will derive an appreciative systems framework. The social process described so far will be overlaid with this framework. This will allow collaborative projects to be viewed explicitly and with greater confidence as dynamic systems. The model building process so far has described the social process using a systems tool which highlights dynamic influences. This phase of the model building makes the case for the social process *as* a system. This chapter corresponds to Step 3 of Schema 1. There is a fold-out of Fig 2 in the appendices.

5.2 OVERVIEW OF APPRECIATIVE SYSTEMS

The systems perspective on social process developed by Vickers is a natural extension of the work of Berger and Luckmann and also of Boulding. Vickers stresses many of the same points but always from a systems perspective. In a sense, the sociology of knowledge developed in chapter 4 provides a common background against which it is possible to sharpen up an appreciative systems foreground. The following quotation is included for two reasons. Firstly it reveals this common background especially in its emphasis on the role of language and conversation, the dialectical process, socialization and normative process, and the creation of social reality. Secondly it shows the systemic foreground with which Vickers is primarily concerned, especially the importance of shared systems of interpretation necessary for meaningful communication which allows stable relations to be formed so that social systems can endure.

> "No communication has any meaning, except in conjunction with the setting of the receiving apparatus; just as a key's significance as a key is related to the lock it is designed to open. But words are in some measure skeleton keys; and in some measure our receiving minds are skeleton locks. Each moulds the other. Talking and being talked to set going in me a circular process which was not only to magnify beyond measure my experience of relations with the world around me but also to define the ways in which I should experience it.

The language I learned contained an implicit order. It had nouns providing categories with which to distinguish objects, events, relations and even the more remote abstractions that I learned to make; adjectives describing their qualities and effects; verbs describing what they could do to me and I to them. These powerfully conditioned the kinds of order which I could distinguish or conceive. With their aid I ordered my own experience in my own particular way.

For these two streams of experience soon began to generate a third. I had to make sense of them, to reduce them to some kind of order. I experienced the inconsistencies inherent in this flood of instruction and interpretation, the conflicts between my own seekings and shunnings, problems arising in the management of fear, of time and of uncertainty. I developed strategies for living; not merely ways of acting but also ways of seeing and valuing which were to define and contain me and which I can revise the less as I learn more clearly to recognize them.

This was a normative process, imposing order on the flood of incoming experience by determining what I should notice, how I should define and value it and what I should feel constrained to do about it. By this process I built for myself the artifactual world in which I live, a world made to measure and equally a world which moulds me. I can compare it with the world of others, in so far as communication enables me to understand differences, as well as to note similarities. But I cannot compare it with some 'real' world, common to all and objectively given, for that is both much more and much less than anybody's world. " (Vickers, 1970, page 74 - 75)

By rejecting a "real world, common to all and objectively given", Vickers is holding a position that is consistent with that of the philosophy of meaning and the social construction of reality.

Vickers regards social systems as ongoing sets or patterns of relations between individuals and organizations. There are two sets of relationships, internal relationships which relate individuals to each other, and external relationships which link a 'continuing entity' to its surrounding environment. Examples of such continuing social entities or social sub-systems might be family, neighbourhood, city, factory, university, or trade union (Vickers, 1968, page 74). Vickers is concerned with the manner in which these sets of relationships seek and attain self-balance and dynamic stability (Vickers, 1968, page 19). Systems will regulate themselves after a fashion (Vickers, 1970, page 127) and will eventually stabilize through automatic limitation (Vickers, 1968, page 22) but most man-made systems require regulation according to standards set by a deliberately formulated policy. The object of policy, regulation and government are to "preserve and increase the relations we value and to exclude or reduce the relations we hate" (Vickers, 1970, page 125). Deliberate regulation involves the formulation of acceptable standards through collective choices, but, because there are usually different ways of seeing the same situation, these choices are multi-valued (Vickers, 1968, page 123). These 'ways of seeing' affect the communication necessary for the whole process of regulating relations according to shared standards derived from collective, multi-valued choices. Vickers believes that there is a communication crisis caused by the breakdown in the shared systems of interpretation which allows communication to have meaning (ibid., page xiii). He calls this shared system of interpretation an appreciative system:

"Professor Boulding (1956) has described our inner view as 'the Image' and he has most usefully stressed its importance and dimensions; but to picture the inner world we must look behind the image and ask what causes an individual to see and value and respond to its system in ways which are characteristic and enduring, yet capable of growth and change. A national ideology, a professional ethic, an individual personality resides not in a particular set of images but in a set of *readinesses* to

see and value and respond to its situation in particular ways. I will call this an appreciative system." (ibid., page 41)

The following quotation elaborates on the concept of an appreciative system and relates it to the communication crisis with which Vickers is so concerned:

"....let us be clear in what this crisis of communication consists. We do not lack means of communication. On the contrary, we have so great a superabundance that the task of selecting, collating, storing, and retrieving outdistances our best efforts to keep pace with it. This, however, is not the crisis I have in mind, though it is indeed critical. I am concerned not with the means of communication but with the shared appreciative system which must interpret it and which alone gives information meaning. ....information is an incomplete concept, developed by communication engineers who could legitimately assume that sender and receiver were linked by a common appreciative system and who were not concerned with the significance of what they transmitted in developing and changing the very system which gave it meaning. This, however, far more than the means of communication, is at the heart of communication science and the heart of the current crisis in communication. For it is this which is in danger of losing its coherence and its continuity; and it is on this that we depend absolutely for our specifically human functioning.

It is very odd that we have no name for these states of readiness to discriminate and to evaluate which are both the product and the condition of human communication - unless, indeed, their name is 'mind'..... I call it an appreciative system." (ibid. page 130 - 131)

Vickers is essentially confirming the limitations of communication as per the conduit metaphor which assumes a shared system of interpretation. He is concerned with meaning, a central theme of this research thesis.

5.3 NOTATATION OF THE APPRECIATIVE SYSTEMS FRAMEWORK

In the summary of Chapter 4, the causal influences of three subsystems relating community, commitment and collaboration, were summarised in a common framework in Fig 11. This showed the similarity in the structure of these subsystems. It should be noted that, whereas the narrative description of Fig 1 in section 4.5 used the language of 'influencing', the language used during the narrative description of Fig 11 in section 4.6 had subtly changed and taken on the flavour of describing a process. The danger is the potential of introducing confusion as to what the diagramatic notation really means. During the description of Fig 1 an influence diagram was used to describe a system of influences inherent in a social process. Fig 11 uses the same notation but its accompanying narrative actually describes a social process directly. If we take this to a logical conclusion, Fig 11 could be reframed as a high level summary of a social process and the language and notation of 'influencing' could be dropped altogether. Fig 12 makes this transition.

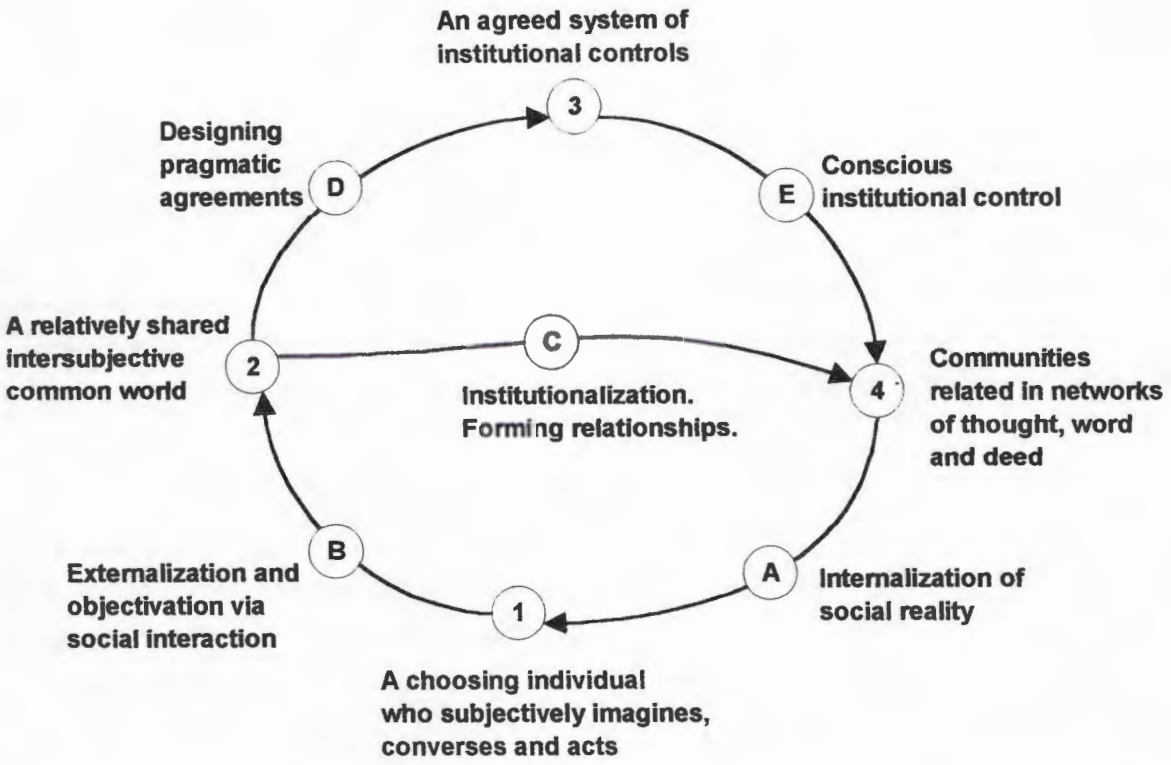


Fig 12. A summary of social process

The intention is to try to incorporate the idea of a system of influences as well as the idea of social process into one model and not to be confused about the notation. Fig 1 has no labelling of influences (other than the direction of the influence) and, however difficult it may be to actually measure them, Fig 1 has influencers which are framed as being measurable. By virtue of the adjectives associated with the nouns, for example 'variety of activities', it is possible to conceive of 'variety' increasing and therefore influencing (in the same direction) an increase in 'variety of forms of life'. The subtle movement away from the language of 'influence' toward the language of 'process' is initiated by labelling the influence, as occurred in Fig 11. Fig 1 is oriented toward describing the *influencers* and limits its discussion of the *influence* itself to just the direction of the influence. When the influence is labelled and then described, as in Fig 11, it promotes discussion as to the nature of the influence itself which in turn promotes discussion about process. This change in emphasis is due to the inherent difference between the language associated with *influencers* and the language associated with *influences*: *influencers* utilise nouns and adjectives to imply a 'measurable state', whereas labelled *influences* use verbs to imply the nature of a 'dynamic movement' between these measurable states.

The above distinctions are important because, in Chapter 6 it will be shown, from a practitioner's point of view, that both perspectives are desirable. The *influencers* give entry points of inquiry into the state of a system, allow one to take stock of a situation and, given a particular state, its likely influence on the whole system. The *influences* allow one to ask a different set of questions relating to the mode of

activity or process that brought the influencers about. Influences introduce a dynamic perspective that enriches systems inquiry.

Fig 12, being a direct description of social process, makes further changes and effectively deprives the model of its influencers. The clusters of measurable influencers in Fig 11 are reduced in Fig 12 to nouns without those adjectives which imply measurability so that measurability is less apparent. Instead of a cluster of measurable influencers, we now have single nouns which are less explicitly measurable and any adjectives which may describe them emphasize descriptive properties rather than measurable states. The labelling of the direction of the influences has also been dropped.

Fig 1, Fig 11 and Fig 12 all provide useful perspectives on the sociology of collaborative projects, but the intention is to provide a framework that encompasses the strengths of all these models. Fig 2 depicts the appreciative systems framework which builds on all three perspectives. The model retains the clusters of influencers but subsumes each cluster under a descriptive noun drawn from the vocabulary of appreciative systems. These nouns, namely Generator, Appreciator, Regulator and Relater, are *influencer types*. Their purpose is to unite the clusters of influencers under a common categorization which expresses the common nature of the clustered influencers. The influences, namely Worldmaking, Forming, Socializing, Designing, and Governing, are also labelled and from now on are called *movements*. This avoids the confusing similarity between the words *influencer* and *influences*. In looking at Fig 2, it would seem that much of the language of influencing has been discarded. The direction of influences has been discarded, as well as some of the adjectives implying measurability of the influencers. The intention, however, is to allow the model to be read in two modes, a process mode and an influence mode, and that each mode will reinforce the other. The influencing mode has been laid out in detail in Fig 1 and is implicit in Fig 2 even if some of the notation has been altered. Thus, if there is any confusion about reading the model in the influencing mode, it can be cleared up by reference to Fig 1. For example, Fig 2 reads: *Generator:* Activities - Worldmaking - *Appreciator:* Forms of Life. If there is confusion about what this means, it can be cleared up by reference to Fig 1 which reads as: *Variety of activities* influences *variety of forms of life* in the *same* direction. The process mode is explicit in the appreciative systems model and will be explained during the remainder of this chapter.

In summary, the appreciative systems framework has as an implicit foundation a system of influencers overlaid with an explicit dynamic systems process. Both modes can be read directly from the model, and each mode reinforces the other. By blending these modes together and reading them simultaneously, more is gained from the model than would be the case if each mode were read separately. The model is built up from influences or movements which are labelled with appropriate verbs. These provide a dynamic perspective that spells out the nature of the movement. The model is also built up from implicit influencers which are measurable in the sense that they can increase or

decrease. These influencers are clustered under an influencer type which represents the common structural position of influencers from the different domains, namely acting, thinking and conversing.

5.4 THE APPRECIATIVE SYSTEMS FRAMEWORK

This section derives and describes Fig 2, the appreciative systems framework. There is a **socializing** movement in which individuals are "claimed" as members of a communicative network (Vickers, 1968, page 130). Socialization equips the individual to function as a member of specific communicating groups (ibid., page xvi) represented in Fig 2 by the influencer type of **relater**. The relater is the system of dynamic relations that binds people together in closely knit, mutually accepting groups capable of co-operative action and effective communication (Vickers, 1970, page 19). Although the individual's conceptual world is culturally shaped through socialization, there is some degree to which this cultural heritage can be later criticized, rejected or reshaped (Vickers, 1968, page xvi). By various thoughts, words and deeds, an individual acts as a **generator** by providing the content or events which comprise the social system. The ability to make individual choices and generate novelty creates a certain inner tension that resolves itself in the search for shared meaning through the movement of **worldmaking** (Goodman, 1978, page 7). Worldmaking involves the experience of sharing thoughts with others. The worldmaking movement promotes a degree of shared outlook upon the world resulting in a relatively shared "set of readinesses to see" situations in particular ways (ibid., page 41). Thus the **appreciator** becomes a chosen system of classifications and patterns of action which determine what we notice and what we ignore in particular situations (Vickers, 1968, page 75). Most importantly, the appreciator provides a shared interpretive system without which communication has no meaning (ibid., page xiii). Vickers makes three statements pointing to the possibility of social systems **forming** naturally. Firstly, "life consists in experiencing relationships" (Vickers, 1970, page 128). Secondly, by weaving a net of mutual expectations, habit is the basic regulator in every society (ibid., page 142). Thirdly, systems will tend to seek and find their own stability (Vickers, 1968, page 19) so that, even in the absence of deliberate regulation, there will be some degree of "automatic limitation" (ibid., page 22). A shared appreciator encourages the forming movement that allows social systems to hang together in a natural manner.

To summarize the process at this point would be to summarize the processes and influences underlying a social system which is not deliberately regulated and which can be called the 'main system' (ibid., page 136). There is an influencer type of relater which is a network of various networks of relationships which link individuals together. There is a socializing movement which culturally shapes the individual. There is an influencer type of generator characterized by an individual who makes active choices. There is a worldmaking movement in which individuals combine to create shared meaning and to make common sense of the world. There is an influencer type of appreciator which is a set of readinesses to see situations in certain specific ways to the exclusion of many

others. There is a forming movement in which various networks of relationships come about as a result of the cooperative action fostered by a shared appreciation of the world.

The remainder of the appreciative systems framework is concerned with deliberate regulative behaviour. This seeks to replace unplanned limitation, at which a system naturally stabilizes, with collectively chosen limitation through deliberate regulation of the system's relationships according to policy standards. Essential to regulative behaviour is the influencer type of **regulator** which comprises the standards, thresholds and settings of the system (ibid., page 76). The regulator provides criteria and norms that allow comparison between mutual expectations of the system and its actual state. The **designing** movement is critical to the process of setting the standards which involves creating mechanisms for making collective choices. The appreciator is seldom sufficiently shared to avoid problems being viewed according to multiple, but nevertheless relevant, schemata (ibid., page 123). Making the necessary multi-valued choices requires co-operative dialogue to design both the mechanisms for making collective choices (ibid., page 59) and also collective agreements sufficiently strong and shared to preserve the coherence of the system (Vickers, 1970, page 89). The ability to compare the relationships of the main system with the designed standards of the regulator allow government of the system. The **governing** movement involves the regulative action necessary to align undesirable relationships manifest in the system with designed settings of the system. Governing the system involves a balancing process aimed at keeping the system's metabolic relations in phase, and also an optimizing process aimed at realizing the combination of relations according to the standard set by the regulator (Vickers, 1968, page 115).

At this point it is possible to summarize the essential characterisics of the control system which regulates the main system. There is a designing movement in which both mechanisms for making collective choices and the collective choices themselves combine in a process of policy formulation. There is an influencer type of regulator which comprises standards and system settings of sufficient coherence to enable deliberate regulation of the system. There is a governing movement which detects disparity between relater and regulator, and triggers the activity necessary to realign the relater to the settings of the regulator.

Having outlined the elements of the appreciative systems framework, it is necessary to emphasize how it works as a dynamic whole.

Unlike mechanical systems in which the regulator or "form-giving element " is separated from the dynamic main system (ibid., page 159), the settings of human systems are generated from within the system itself as part of its ongoing dynamic functioning. Individuals learn from their actions. Human societies are capable of innovation and accumulative learning. Essentially, appreciative systems are learning systems characterized by a circular process in which the settings of the appreciative system

are constantly modified by their own exercise (ibid., page 6). Making judgements changes the settings:

".... the schemata on which reality judgements are based are in constant development under three main pressures - the pressure of the event; the pressure of other men's schemata; and the pressure of their own internal requirements. Events call constantly for new appreciations of the 'situation'. Other people's communications reveal schemata which confirm or challenge our own. And, apart from both of these, the inner inconsistencies and incompleteness of our own schemata call us constantly to revise them." (ibid., page 150)

These pressures ensure that the relations in the system are always dynamic and therefore regulation is needed to preserve coherence in the face of change. Stability, rather than being the enemy of change, is a necessary condition for change. Instability occurs when order dissolves as a result of the system generating its own negation too rapidly thus making regulation impossible (Vickers, 1970, page 127). If the shared system of interpretation breaks down under these pressures, communication loses meaning. Because there is no influencer or movement that is unaffected by the quality of the communication, a breakdown of the interpretation system influences breakdown of the entire system. When this happens, the system may still be able to attain some kind of automatic limitation, but this will be unregulated and probably undesirable

Vickers emphasises value systems. Humans "seek and shun" in terms of an inner world that is structured fundamentally by values (Vickers, 1968, page 37) and, through worldmaking, the appreciator becomes value structured. The designing movement is fundamentally concerned with making value judgements, but because values are seldom completely shared, generating sufficient agreement on policy hinges on the way problems are structured. The task of defining a problem is complicated by competing value structures, and ways of seeing the problem are crucial to its solution (ibid., page 83):

"This need constantly to restructure problems makes novel demands on communication. For policy-making is a collective activity and the first condition of communication which makes it possible is that the participants should be talking about the same thing, or at least know when this is not so. Most of the discussion which goes into policy-making is directed to reaching agreement on how the situation can most usefully be regarded; in other words what is the complex of relations most significantly involved. Policy-making is vastly complicated when this cannot be taken for granted but must constantly be reviewed." (ibid., page 84)

A common language is therefore essential to policy formulation. The following quotation emphasizes its creative aspect:

"The situation to which a policy maker attends is not a datum but a construct, a mental artifact, a collective work of art. It has to be simplified or it becomes unmanageable; yet if it is over-simplified, it will be no guide to action. It has to reflect present and future reality; yet if it departs too sharply from the familiar thinking of the past, it will not be sufficiently shared by those for whom it has to provide a common basis for discussion. It has to be not merely discovered but invented, not merely invented but chosen from among several alternative inventions, each a valid but differently selected view. Most difficult of all, it must not obscure the views which it supersedes" (ibid., page 85).

Vickers makes an explicit distinction between "appreciative behaviour" and "regulative behaviour" (ibid., page 137) in self-controlled systems. A simple view of such systems is that there is an ongoing informational process in which signals are generated to which the main systems responds. A controlling subsystem compares information about the state of the main system, especially its relations, with standards or norms set in the regulator. Any disparity generates a signal to which the main system responds (ibid., page 136). In terms of this view, three issues need to be considered:

1. How does the control system derive its information about the main system?
2. How does the control system derive its norms?
3. How does the signal generated cause selection and initiation of change in the main system?
(ibid., page 136)

In answering these questions, Vickers also underlines his contention that regulative behaviour is "norm-seeking" as opposed to "goal-seeking" (ibid., page 135):

"The first and second fields of inquiry - the observation of the 'actual' and its comparison with the 'norm' - are indissolubly connected and important in their own right. The combined process I call appreciation. The third field - the choice of action - is separable and may be irrelevant. Appreciation may or may not call for - and if it does, it may not evoke - action which may or may not abate an observed discrepancy, action which I will call regulative action" (ibid., page 137)

The next section provides a brief summary of the appreciative systems framework and its relation to collaborative projects.

5.5 THE APPRECIATIVE SYSTEM AND COLLABORATIVE PROJECTS

It is convenient to view the system from the perspective of the appreciator. The appreciator has a direct relationship with, a 'clear view' of, all the other influencer types. In this sense it is convenient to regard it as the main driver in the system. From its vantage point, it sees how it was created by the movement of worldmaking. It is in a position to see that portion of the vast variety of activities, images, conversations generated by individuals that was included in the movement of worldmaking. The appreciator was created in this movement and therefore is in a position to reflect on itself and the process by which it was generated. How much of the generator was it possible to see? How much of the generator was filtered out by the current readinesses of the appreciator to notice only certain things? What was the form of worldmaking? Was it conducive to creating shared meaning? Did the nature of the movement itself limit the possibilities? How shared is the appreciator? Is it in a position to act as a sound influencer for designing a regulator or forming a relater? How coherent are the forms of life? To what extent is there a shared public image? Is the mode of discourse shared to the extent that collaboration is possible? Can the appreciator sustain a coherent system?

The appreciator has a view of the relater and the movement which formed it. Similarly it has a view of the regulator and the designing movement. It is in a position to reflect upon the relations which it helped to form and regulative settings whose design it influenced. It can reflect on the ability of the relater to be governed by the regulator because it is in a position to appreciate the disparity between the two influencer types. It could, for example, detect a disparity between purpose and community. Knowing the shared public image which formed community, perhaps high quality products, it could observe that the designed purpose attempts to govern the community according to conflicting settings, such as short term profit. It should be able to reflect on where and how the discrepancy occurred. Maybe it occurred 'at source' in the sense that the appreciator was fragmented. Maybe it occurred because the mechanisms for making collective choices were inappropriately designed.

In a slightly different manner, the appreciator can make inferences about the socialization movement. It sees the relater which forms from it, and it experiences the product of socialization through the worldmaking movement. The success of socialization can be assessed by the amount of change generated by individuals. A small amount of change allows the appreciator to function, but if there is too much change, the system may dissolve into instability. The appreciater may resist variety by holding on to its readinesses to notice only certain things or by employing a worldmaking movement that resists change.

It is important to see the appreciative systems framework as a whole. The subsystems are not discrete, separable components. They are different perspectives of the same phenomenon. In much the same way that a landscape may be drawn, photographed or mapped, collaborative projects can be represented as communities, commitment networks or collaboration networks. Drawings, photographs and maps are in some respects related and in some respects not related. So too are the perspectives of community, commitment and collaboration. The interrelationships and overlaps are of interest. Using a map and a photograph gives a fuller perspective than either on its own, but it is necessary to know how to relate the two in order to get this fuller perspective. The landscape remains a landscape, however, and it is only our perspective which changes. The next chapter completes the model building and provides the finalized lens through which to view collaborative projects.

5.6 JUSTIFYING 5 KEYWORDS AS INTRA-SUBSYSTEM MOVEMENT NAMES

Section 4.5 introduced 14 keywords but pointed out that they could not be fully justified at that stage because full justification would in some cases depend on theory still to be developed in later chapters. Now that the appreciative systems framework has been derived, it is possible to justify the 5 keywords associated with influences *within* subsystems as intra-subsystem movements. Section 5.4 has included the keywords of socializing, worldmaking, forming, designing and governing as movements names. The following subsections justify the choice of names by attributing them to specific sources.

5.6.1 Socializing

This word is appropriated directly from Berger and Luckmann's account of the sociology of knowledge as described in subsections 4.4.13, 4.4.14 and 4.4.15. Vickers also emphasis socializing as detailed in section 5.4

5.6.2 Worldmaking

This word is appropriated from Goodman who argues from a pluralist position that a "multiplicity of worlds" is made from multiple frames of reference:

"Consider, to begin with, the statements "The sun always moves" and "The sun never moves" which, though equally true, are at odds with each other. Shall we say, then, that they describe different worlds, and indeed that there are as many different worlds as there are mutually exclusive truths? Rather, we are inclined to regard the two strings of words not as complete statements with truth-values of their own but as elliptical for some such statement as "Under frame of reference A, the sun always moves" and "Under frame of reference B, the sun never moves" - statements that may both be true of the same world.

Frames of reference, though, seem to belong less to what is described than to systems of description: and each of the two statements relates what is described to such a system. If I ask about the world, you can offer to tell me how it is under one or more frames of reference: but if I insist that you tell me how it is apart from all frames, what can you say? We are confined to ways of describing whatever is described. Our universe, so to speak, consists of these ways rather than of a world or of worlds." (Goodman, 1978, page 2 - 3)

In other words, the construction of reality depends on "ways of worldmaking" (ibid., page 7). Goodman argues that worlds are made from other worlds:

"Worldmaking as we know it always starts from worlds already on hand; the making is a remaking." (bid., page 6)

Subsection 3.3.5 introduced metaphoric redescription as an additional tool for the designer of collaborative projects. Goodman emphasises metaphoric ways of worldmaking:

".....worlds are made not only by what is said literally but also what is said metaphorically, and not only by what is said either literally or metaphorically but also by what is exemplified and expressed - by what is shown as well as by what is said." (bid., page 18)

Worldmaking emphasizes that a collaborative project needs to construct its own reality, that this involves remaking the world, that it can be remade metaphorically as well as literally and that the frame of reference or system of description will determine this remaking and therefor the degree to which individuals have a shared appreciation of the world.

5.6.3 Forming

This word is appropriated from Churchman and has already been covered in subsection 3.3.1. The two following quotations add support for the choice of the word:

"Politics means primarily the way in which people gather around issues of human living, food, shelter, education, patriotism, war, security, etc. Each such gathering together I call 'forming polis'." (Churchman, 1979, page 24)

"...the 'political approach' to the management of human affairs is 'making polis (or community).' One of the essential conditions for making polis is the creation of an image of sharing and a breakdown of the ego-desire" (bid., page 74)

The way of worldmaking will determine whether there is a genuine sharing or just an illusion of sharing. The designer will always be involved with the politics of the collaborative community so Churchman's notion of 'forming polis' is a better choice of movement name than 'institutionalizing' which could have been appropriated directly from the vocabulary of the sociology of knowledge.

5.6.4 Designing

Section 5.2 discussed Vickers' conception of appreciative systems which emphasizes deliberate regulation of social systems according to standards set by a deliberately formulated policy. Subsection 3.3.1 discussed measures of system performance in Churchman's conception of the design of purposeful systems. The sociology of knowledge, subsection 4.4.10, discussed the need for additional systems of institutional control to supplement primary institutional control. The word 'designing' is chosen to represent all these perspectives on the need to supplement the natural formation of community by deliberate regulation, involving the designing of standards, policy, institutional controls, purpose and measures of performance.

5.6.5 Governing

This word is appropriated directly from Vickers' conception of appreciative systems and has been adequately covered in section 5.4.

CHAPTER 6. A MODEL FOR THE DESIGN OF COLLABORATIVE PROJECTS

6.1 DERIVATION OF THE CPD MODEL

The appreciative systems framework depicted by Fig 2 effectively incorporates 15 of the 24 influences contained in Fig 1. Influences 4 through 18 of Fig 1 are catered for by a common framework first described in section 4.6 and then overlayed with an appreciative systems framework in section 5.4. The 5 shared keywords were justified as intra-subsystem movements in section 5.6. Effectively, the derivation of the three sub-systems of CPD has been completed. All that remains to derive the full CPD model is to incorporate the remaining 9 influences which act *between* the subsystems. In order to do this, the three subsystems which were viewed within a single framework in Fig 2 must be re-imposed on Fig 1, the full original causal influence diagram. This requires that each subsystem is once again to be viewed separately, but at the same time the notation of the appreciative systems framework (developed in section 5.3) must be preserved and the influencer types and intra-subsystem movement names (developed in sections 5.4 and 5.6) must be overlayed onto Fig 1. Having performed this transformation, the final CPD model will be complete in all respects except one - the 9 influences *between* subsystems and their associated keywords need to be justified as inter-subsystem movements. Schema 1 shows that this is the main task of Step 4 and Chapter 6. There is a fold-out of Fig 3 in the appendices.

In summary, CPD is derived from three sources:

1. CPD is derived from practice in practical situations. This practice is summarised in a causal influence diagram, Fig 1. The social process inherent in Fig 1 is supported by the theory of the sociology of knowledge. CPD represents the dynamics of the social construction of the reality of a collaborative project.

2. CPD is derived from theoretical research into the nature of shared meaning. Table 1 summarizes this and provides a set of distinctions and insights to guide the designer when using CPD.

3. CPD is derived from systems theory. The appreciative systems framework, Fig 2, provides a systemic lens to organise the designer's mind.

6.2 JUSTIFYING 9 KEYWORDS AS INTER-SUBSYSTEM MOVEMENT NAMES

6.2.1 Explicating, performing, trusting and coordinating

These inter-subsystem movements can be discussed jointly by considering the manner in which they interact to influence "trust in commitment":

"We use language in human activities, and our use of linguistic forms is shaped by the need for effective coordination of action with others. If one person's utterance is not intelligible to others, or its interpretation by the listener is not consistent with the actions the speaker anticipates, there will be breakdown..... it results in the loss of trust in commitment. If I say there is water in the refrigerator and this assertion is not consistent with the domain of relevant actions, you may decide that you can't "take me seriously" or "believe what I say." A fundamental condition of successful communication is lost." (Winograd and Flores, 1986, page 62 - 63)

This supports Vicker's conception of communication and interpretation as outlined in sections 5.2 and 5.4. The interpretation of this quotation has implications for a large section of CPD. It justifies the coordinating movement. Furthermore, it links coordinated action to a shared system of interpretation. What is explicated and what is performed need to be consistent otherwise there is a general loss of trust in commitment and communication throughout the system. As the projects of Chapter 7 will show, interventions in the design of collaborative projects focus on establishing a shared mode of discourse, in other words, a shared interpretative system. If this can be achieved, then two influences can be enabled. Firstly, through the design of synchronization mechanisms, activity can be coordinated. Secondly, a collaboration network can form which has the capability to explicate clear and unambiguous commitments. If trust can be engendered by successful performance of these commitments, then the relationships within the collaboration network, the commitment network and community are strengthened.

The four movement names 'explicating', 'performing', 'trusting' and 'coordinating', are not appropriated from a single source, but derived from the sociology of knowledge of Berger and Luckmann, the appreciative systems of Vickers, the philosophy of Winograd and Flores and findings from action research.

6.2.2 Expressing and representing

Section 4.4.26 highlights the importance of conversation, especially that "men must talk about themselves until they know themselves" (Berger and Luckmann, 1966, page 53). This suggests that an ability to converse is integral to an ability to form private images. Dialogue and 'socio-therapy' are concerned with the assumptions underlying thought. The process of unfolding private images, suspending assumptions and creating a shared public image necessitates that private images are expressed in conversation that is meaningful not only for others, but also for oneself.

The discussion on worldmaking, subsection 5.6.2, highlighted "what is exemplified and expressed" (Goodman, 1978, page 18). Expression and exemplification are integral to representation. The interventions described in Chapter 7 highlight the importance of representing shared meaning and off-loading it as shared space which can then become the basis for designing a shared mode of discourse. Often the process of 'turning' communication into collaboration hinges on this movement. Goodman emphasises non-verbal and non-literal modes of representation:

"....attention usually focuses on versions that are literal, denotational, and verbal. While that covers some - though I think far from all - scientific and quasi-scientific worldmaking, it leaves out perceptual and pictorial versions and all figurative and exemplificational means and all nonverbal media. The worlds of fiction, poetry, painting, music, dance, and the other arts are built largely by such nonliteral devices as metaphor, by such nondenotational means as exemplification and expression, and often by the use of pictures or sounds or gestures or other symbols of nonlinguistic systems. Such worldmaking and such versions are my primary concern here; for a major thesis of this book is that the arts must be taken no less seriously than the sciences as modes of discovery, creation, and enlargement of knowledge in the broad advancement of understanding, and thus the philosophy of art should be conceived as an integral part of metaphysics and epistemology." (bid., page 102)

Designing appropriate modes of representation is the most critical and most difficult aspect of most of the interventions recorded in Chapter 7. This is the area of collaborative project design where project members need most assistance. The verbal, literal, denotational modes of discourse are familiar and comfortable to most project members, but these modes are not always suitable to exemplify and express shared meaning. 'Looseness' of shared meaning often becomes apparent only when one attempts to explicitly represent supposedly shared public images according to an agreed mode of representation. In other words, the verbal, literal, denotational mode is often inherently 'looser' than some of the other forms that Goodman suggest above. For example, two people might be able to agree on a verbal design of a house, but disagree on each other's representations which uses standard building plans. Building plans are a more exacting mode of representation and force cohesion which can be overlooked in purely verbal discussion. My contention is that the journey from 'loose complexity' to 'cohesive simplicity' is achieved by off-loading aspects of a world made in common via an appropriate mode of representation that exemplifies those aspects of the common world.

6.2.3 Transforming

Lakoff and Johnson contend that "people in power get to impose their metaphors", and that new metaphors have the power to define new realities (Lakoff and Johnson, 1980, page 157). As has been discussed in subsection 3.3.4, they also contend that our conceptual systems are structured by metaphors and this is reflected in our language (bid., page 3). Language in turn governs the commitment network. The implication of this is that new metaphors, especially 'deep metaphors', create new realities thereby restructuring the conceptual system and creating new languages. If this is aligned to purpose, the language subsystem can be transformed and regulated by the redescribed metaphor subsystem.

6.2.4 Legitimizing

This word is appropriated directly from Berger and Luckmann's account of the sociology of knowledge as described in subsection 4.4.20.

6.2.5 Normalizing

Subsection 3.2.1 argued that a form of life is an activity or collection of related activities of a certain kind which have become regularized to some degree. This implies that certain shared public images are normalized in connection with this regularized form of life.

6.3 THE STRUCTURE OF CPD

It will be noted that influences 1, 2 and 3 of Fig 1 act on the same influencer type in Fig 3, the relaters. Similarly, influences 19, 20 and 22 act on appreciators. Influence 21 acts on regulators and influence 23 acts on generators. Only one influence, influence 24, acts on different influencer types, in this case regulator and generator. Thus within subsystems, movements act on different influencer types, but between subsystems, movements act on the same influencer type with only one exception. This phenomenon was not consciously designed into CPD, but has come about through evolutionary testing of Fig 1 in practice and overlaying it with an appreciative systems framework. Fig 1 was not tampered with to provide CPD with a certain pattern, but nevertheless, some patterns are apparent and are useful to the practitioner. For example, six of the nine inter-subsystem movements move in a clockwise direction so that each subsystem influences another subsystem by only one movement in a counter-clockwise direction. Also, no influencer is influenced by fewer than one or more than three movements.

It should also be noted that there are only six influences in Fig 1 that act in an opposite direction (e.g as influencer A *increases*, influencer B *decreases*). These six influencers are all inter-subsystem influences and emanate from the language subsystem or act back on it. This is significant for the next section.

6.4 USING CPD IN PRACTICE

This section can be seen as a continuation of section 5.5 which related the appreciative systems framework to collaborative projects.

The initial intention at the beginning of the research process was to develop a methodology that would specify a sequence of steps and practices to be followed during the design of collaborative projects. This was later abandoned for a number of reasons. As will be seen in chapter 7, there are many situations where interventions are necessary, but where it is also impractical to use an explicit methodology for a number of reasons. For example, one may be called upon to intervene in a project that is already failing. Such a project will not benefit from a methodology geared to designing new collaborative projects from scratch. A project member with collaborative insight may be more effective working discretely in the background and may have no political power to implement an explicit methodology. It became increasingly apparent as the research progressed that a powerful organising

lens and a rich set of insights can enable the successful design of situation-specific interventions. Chapter 7 details a number of successful interventions based not on a methodology, but on an increasingly richer lens and set of insights. It seemed useful to carry on in this vein, so instead of developing an explicit methodology attention was focused on the lens, Fig 3, and the set of insights, Table 1.

This makes it difficult to describe how to use CPD in practice. In the absence of a rigorous methodology, the practitioner needs to develop collaborative skills. Section 3.5, which described the ACT Movement project, illustrates that the collaborative skills are transferable, and able to be learned and effectively used by newcomers. This project was 'turned around' by the individual who could not cope initially. No special techniques or methods were necessary other than an ability to 'see' the poor design of the initial project conversation. The individual then designed the interventions to counter the poor design. CPD provides the lens through which to 'see' the design of a project.

The influencers and movements focus attention on certain aspects of a project. In effect, each influencer and movement prompts the designer to ask questions about the design of the project. For example, it will be seen in Chapter 7 that in some projects a major problem is that there is no shared mode of discourse, while in others this is no problem at all. The former require interventions to establish a shared mode of discourse, whereas the latter are concerned with using a shared mode of discourse to design effective synchronization mechanisms or to clarify shared purpose. CPD does not provide a set list of questions. Instead, it provides a systemic representation of the social processes and influences operating in *all* collaborative projects, and invites the designer to assess the viability of a particular project in these terms.

The purpose of CPD is to sensitize the designer to notice certain aspects of a project. Because CPD represents a project as a system, it encourages the designer to intervene in such a way as to contribute to the effective functioning of the overall system. This requires additional insights from Table 1. The different subsystems of CPD focus attention on different domains of collaborative projects. Table 1 summarizes critical distinctions that the designer should bear in mind when 'seeing' projects and designing interventions.

I will describe the use of CPD in terms of the research problem as it was initially raised in Chapter 2. The software process was described as a journey from 'loose complexity' to 'cohesive simplicity'. The language susbsytem represents 'loose complexity' and the conversation subsystem represents 'cohesive simplicity'. The metaphor subsystem acts as a bridge between the two and may be called 'shared meaning'. Looking back through the lens to Fig 1, influence 5 asserts that an increase in the variety of activities influences and increase in variety of forms of life. This in turn generates increasing complexity throughout the language subsystem. And, because the language subsystem influences the other two subsystems, as the language subsystem grows in complexity, it influences the whole

system to become complex. All red movements in CPD emanate from the language subsystem, and the designer can look to these movements for sources of loose complexity. A complex and loose language subsystem produces a fragmented commitment network which makes it difficult to perform adequately and thus undermines the cohesiveness of community. An increasing variety of forms of life lessens the chances of establishing a single shared mode of discourse and the public images are likely to be fragmented.

The designer's struggle will be to influence the language subsystem to become simpler and more cohesive through three counter movements that act back on the language subsystem. These are the 'coordinating' and 'explicating' movements of the conversation subsystem, and the 'transforming' movement of the metaphor subsystem. The language subsystem becomes unacceptably loose and complex when it is no longer possible to communicate meaningfully. The designer will then attempt to build shared meaning through dialogue. Shared meaning can itself act back on the language subsystem by transforming the language game. Shared meaning can also act as a foundation for collaborative conversation required by the conversation subsystem. In short, the green movements of the metaphor subsystem spread shared meaning throughout the system. Shared meaning is a stepping stone to cohesive simplicity.

The conversation subsystem is founded on explicit representations and trust built through shared meaning, and is concerned with designing viable collaborative conversations to limit the variety of the language system. Extremely important is that the regulator of the conversation subsystem, which is concerned with simplicity and cohesion, influences the generator of the language subsystem which is the source of looseness and complexity. Whereas the metaphor subsystem has the potential to transform the language system, the conversation subsystem has the potential to limit the variety of the language subsystem. The blue movements of the conversation subsystem spread cohesive simplicity through the system.

Table 1 provides insights as to how this journey from loose complexity to cohesive simplicity might be accomplished. The designer needs to enable the three prototypical conversation types that are appropriate for each subsystem. The language subsystem will be more effective if individuals work conscientiously toward preserving a meaningful interpretative system. Communication as per the 'world re-maker' metaphor may enable this. If the interpretative system breaks down, it may be partly rebuilt through dialogue as per the 'socio-therapy' metaphor. The shared meaning generated by dialogue may be sufficient to transform the language game of the language subsystem. But it may be necessary to off-load this shared meaning into shared space in the form of a manipulable representation and to design a collaborative conversation. The designer will engage in 'conversation choreography' to design a synchronized collaboration network.

In practice, Table 1 acts as a guide for the designer, and the conversation types are prototypes that are often not perfectly achievable in practical situations. For example, as will be shown in Chapter 7, there are many situations where it is critical to build shared meaning but where it is not possible because of time pressures and political restraints to conduct a prototypical group dialogue as described in sub-section 3.3.3. It might be impractical, for example, to get all the participants together in the same room at the same time. Table 1 does, however, provide a cluster of insights to help design a conversation suitable for the specific situation. The conversation may not be a perfect dialogue, but the designer can still attend to the principles outlined in Table 1. The designer will be conscious of the need to unfold background and create multiple representations of the situation. The conversation may 'sweep in' multiple perspectives or metaphorically redescribe the situation, thus challenging assumptions and creating new images or metaphors. Learning may be achieved, humiliation and cruelty avoided, and a collective group intelligence unfolded even if people are not in face-to-face conversation. If the designer is consciously trying to design for collaboration, she may design the conversation so that the shared meaning and associated mental representations are easy to off-load and therefore suitable as shared space for a collaborative conversation.

Chapter 7 will provide additional insights as to how CPD can be used in practical situations. In summary of this chapter, the following gives a consolidated set of guidelines on how to use CPD:

1. Fig 3 organises the mind of the practitioner and thus guides inquiry into the real world situation. It gives the practitioner 36 influences or movements to look for and assess in terms of the system dynamics operating in the real world collaboration. The focus is on "seeing" and framing the dynamics currently operating in the collaboration.

2. Assess the systems dynamics of the collaboration by identifying those particular influences which are operating in such a manner as to inhibit collaboration. Identify those influences which are most inhibitive. Identify how the influences combine to mutually inhibit collaboration. This process identifies where interventions should take place in order to promote more effective collaboration.

3. The interventions need to be designed. This is supported by insights from Table 1. Identify from which sub-system the inhibitive influences emanate and relate it to the corresponding set of insights provided by Table 1. For example, influences emanating from the metaphor sub-system (green influences) require insights from the domain of acting in Table 1 and thus a conversation design of dialogue. Counter inhibitive influences through re-designing the current conversations and use the principles and insights offered by Table 1 as the basis of the conversation design.

4. Repeat steps 1 to 3 iteratively in order to manage the unfolding of the collaboration and to continually learn and reflect on the dynamics of the system being managed and also the effectiveness of the designed interventions.

CHAPTER 7. ACTION RESEARCH

Section 2.11 described the research methodology. It emphasized the importance of both literature research and action research to the evolution of the causal influence diagram (Fig 1) upon which CPD was built. This chapter describes the contribution of action research projects to the evolution of CPD and also describes the testing of CPD in practice.

Each section in this chapter corresponds to an action research project. Each section will include subsections which describe the action research project in order to show how each project contributed toward the evolution and testing of CPD. It will also include subsections which provide a descriptive background to each project. Additional subsections will be included to frame or analyse a project explicitly in CPD terms. These latter subsections will highlight (by using **bold type** in the text) both the use and usefulness of the vocabulary of collaborative projects. The text will explicitly and intentionally incorporate the CPD terminology of Fig 3 and Table 1. The fold-outs of these models in the appendices should be used to assist in relating the phrases in bold type with the terminology of the models. In order to clearly link CPD terminology to the framing and analysis of the interventions, the writing style of these subsections may seem somewhat clumsy and contrived at times, but it is done to clarify the link between the theoretical framework and the action research. Early projects were completed before CPD had evolved to its final form and therefor the analysis of these projects in CPD terms is partly retrospective.

7.1 PROJECT 1 - THE ACT MOVEMENT PROJECT

The background and intervention of the ACT movement project has been described in section 3.5. This section gives an analysis of the intervention in CPD terms.

The project crisis was characterized by an inability to hold **viable conversations**. The analyst simply could not understand the Vantage **language game** which, through the **legitimizing** movement, had become *innapropriately* established as the **shared mode of discourse** for the *whole* project. The **language game** and **private methods** associated with the systems analysis **form of life** provided no help in **governing** the **commitment network**. This caused the analyst to lose the ability to influence the project that she was supposed to be managing. The **commitment network** consisted of little more that an agreement to hold a series of poorly synchronized meetings to 'talk about movements and VANTAGE transactions'.

The analyst was urged to consider the **conversation design** of her project. **Metaphoric redescription** of the project as a 'conversation' rather than a 'production line' influenced the **private images** of the analyst and therefore her approach to designing the project. In particular, she became aware of the manner in which the '**conduit metaphor**', which dominated **communication**, was

negatively influencing the **conversation design**. The **commitment network** needed to be **governed** by a movements **language game** in order to counteract the current **conversation design** which consisted of a one-way transmission of the entire body of VANTAGE transaction information. This undermined **viable conversation** because no shared **interpretative system** existed that allowed the analyst nor the actuaries to create meaning from these transmissions. A movements **language game** needed to be created in order to become part of **the shared mode of discourse** and to counter the existing **legitimizing** movement which promoted only the VANTAGE **language game**.

The analyst initiated a process which influenced the creation of the new movements **language game**. Through a **worldmaking** movement emphasising **dialogue** and **group learning**, a **shared public image** was unfolded from multiple **private images**. This was **off-loaded** as an explicit representation which exemplified the **shared meaning** of the word 'movement'. Through **observation** of the language that the participants used when they talked about movements, in other words by observing **languaging**, the new movement **language game** was extended into a more detailed representation. The analyst noted the distinctions which the actuaries made when talking about movements and used them to design a **mode of representation** using a spreadsheet **metaphor**. The spreadsheet, called the 'movements grid', represented the categories and distinctions of the movement **language game**. Through the **representing** movement, this **off-loaded representation** became **shared space** for a sub-collaboration involving the actuaries. The actuaries were invited to manipulate both the **mode of representation** and its contents in order to align it to their **shared public image** of movements. By influencing the movements grid to become a **manipulable representation** in **shared space**, the analyst had created a **shared mode of discourse** that enabled the **forming** of a **collaboration network**. In other words, **conversation** focused on the representation and how to manipulate it to become appropriate for the movements project. Subsequent **expressing** of **private images** was achieved by using this **shared space** as the common medium. The **shared meaning** of the metaphor **subsystem** was converted into **cohesive simplicity** due to the fact that **representing** a **shared public image** as an **off-loaded representation** is a very exacting movement. The **off-loaded representations** were explicit, tangible and left little room for ambiguous reinterpretations. The **cohesive simplicity** within the **conversation** subsystem influenced the **commitment network** to become more cohesive through the **explicating** movement. The **speech acts** involved in **forming** the **commitment network** were now detailed, precise and **governed** by a meaningful language game. The movements grid *committed* the participants to a *precise* language game. The **commitment network** was now **formed** from a mutual commitment by the actuaries, VANTAGE kingpin and analyst to complete the spreadsheet as a **group collaboration** in **shared space**.

For example, an empty cell in the spreadsheet committed the actuaries, analyst or VANTAGE kingpin to specific **activities**, namely deriving content for that cell. Furthermore, these **activities** would be **governed** by categories consistent with the movement **language game** because the row or column

headings of each cell reflected meaningful expressions in the movements vocabulary. In this manner, the spreadsheet was an effective **synchronisation mechanism** capable of **co-ordinating activity** and an effective measure of the progress and state of the project. The former **loose complexity** resulting from **communication** as per the **conduit metaphor**, had become **cohesive simplicity** as a result of **collaboration** as per the **conversation choreography** metaphor.

In a similar manner, the analyst designed a **collaboration** with the Vantage expert. This involved creating **shared space** by designing an appropriate **mode of representation** which exemplified the important distinctions of VANTAGE transactions which were significant for actuarial movements. The VANTAGE kingpin was invited to correct this **manipulable representation** and in doing so it became **legitimized** as the **shared mode of discourse** for another sub-collaboration.

Thus the **forming** of the **collaboration network** involved two sub-collaborations. A **synchronisation mechanism** was created by **designing** a learning mechanism that enabled the actuaries' sub-collaboration to influence the VANTAGE sub-collaboration. **Learning** from one sub-collaboration was mapped onto the **manipulable representation** of the other sub-collaboration through the **representing** movement. Thus the actuaries manipulated their movement grid as they learnt about VANTAGE transactions, and they also mapped corresponding movements onto the VANTAGE transaction models. In this way both sub-collaborations were involved in a viable **learning** process involving a **shared mode of discourse** and a shared **intepretative system**. The movements **language game** was legitimately involved in **the shared mode of discourse**. The **conversation design** had been countered so that the **conduit metaphor** no longer dominated an ineffectual **communication**. Instead, there was now a **collaboration** which was **governed** by a well designed **synchronisation mechanism** geared toward **group learning**. The **synchronisation mechanism co-ordinated activity** until the end of the project.

7.2 PROJECT 2 - THE CORE/SATELLITE 'CRISIS'

7.2.1 Background

This project took place in a South African asset management company that was in the midst of chaotic change. The asset management company was a subsidiary of a large life assurance company, and was undergoing radical transformation necessitated by the fact that it was under-performing in both its investment performance and its administration services. It was an embarrassment to its parent company not only because it consistently ranked in the last quarter of the industry in terms of investment performance, but also the administration of the assets and investment portfolios was error-prone and poorly managed. This was exacerbated by the haphazard implementation of an administration software package called IMPART.

I will call this project "Core/Satellite" for convenience. At the time that this project took place, March 1997, the following upheavals and change were being experienced:

1. A new managing director had been appointed about six months earlier. His brief was to transform the company and act as chief investment officer.
2. A new administration manager had been appointed about four months earlier. He was a director whose brief was to transform the administration function.
3. A new I.T. manager had been appointed about three months earlier. He had just initiated a plan to reform the I.T. area.
4. A firm of consultants had been contracted to define and implement a new investment strategy. I will call these consultants the "Investment Consultants", and the investment strategy that they came up with came to be known as "Core/Satellite".
5. A different firm of consultants had been contracted to assist in transforming the administration operation. I will call these the "Admin Consultants".
6. Not only was there animosity between the investment function (on the 4th floor) and the administration function (on the 3rd floor), but this spilled over into a professional rivalry between the two consulting firms. Each area tended to blame the other for poor performance.
7. IMPART, the administration package, was new and poorly implemented. A great deal of confusion reigned in administration because critical information was held in the head of two kingpins who reported to the I.T. manager. These kingpins were protective of their knowledge and not inclined to pass it on to the new recruits. This also helped them to conceal mistakes and 'quick fixes' made during implementation of IMPART. They were bottlenecks to both administration and I.T.
8. Existing staff were demoralized and tired. Each inefficiency caused more work and criticism. With a totally new management team, two external consulting firms and a new administration system, it was difficult to cope with all the change and to improve individual performance.
9. I had been seconded to the asset management company to help facilitate the changes and to act as a systems analyst. I reported to the I.T. manager.

The Core/Satellite crisis was sparked by a number of events. The Investment Consultants in conjunction with the managing director announced that the Core/Satellite investment strategy would be implemented on 1st May 1997. The administration function did not know at this stage what these words meant. The Admin Consultants were in the process of planning an "admin transformation", but this would have to be put on hold while everyone scrambled to meet what seemed like an unachievable deadline. Particularly worrying was the fact that the existing administration function had not yet been stabilised on IMPART and would have to be re-organised to cope with Core/Satellite. The worry about IMPART and I.T. in general prompted the managing director, without consulting the administration management or I.T. management, to call in external I.T. consultants affiliated to the Investment Consultants. Their brief was to satisfy the managing director that I.T. could cope with

Core/Satellite implementation. The arrival of this third set of new, external consultants angered the administration management, I.T. and the Admin consultants. It added yet another set of new people to an already confused and tense situation.

The crisis became explicit in a meeting run by the Admin Consultants which was intended to produce a plan to cope with Core/Satellite. The meeting, which I will call the "acrimonious meeting", was attended by all managers, middle managers and consultants from the administration area. It became obvious that the Admin Consultants had no concrete plan, had not consulted internal experts who knew how the business operated, and had no support from the employees who worked permanently for the company. Administration management knew that they were in trouble in terms of the May 1st deadline, but had no alternative plan as to how to deal with the situation. This was all being witnessed by the external I.T. consultants who could not gain reassurance from the process. A meeting was set for the following afternoon but it was clear that the new management team and the Admin Consultants did not know how to cope. Rapid implementation of Core/Satellite was absolutely critical to the continued existence of the asset management company.

7.2.2 Framing the intervention

CPD and Table 1, in combination, provide a lens to assist in framing the situation and a means for organising the mind of the project designer. The situation called for a collaborative project, an 'act of shared creation', because it required the administration management team to create a viable plan of co-ordinated activity to respond to the Core/Satellite initiative. In terms of CPD, this relates directly to the language subsystem's need for **activities** to be synchronised through the **co-ordinating** movement and for a **commitment network** to be agreed through the **explicating** movement.

CPD helps to understand the influences causing the 'acrimonious meeting' to fail. There was no regularised **form of life** as a result of three new transformation initiatives, a completely new set of management personnel (including three sets of consultants) and the new administration system (IMPART). The **activity** influencer was generating too much variety and the **worldmaking** movement was unable to limit this variety in order to establish a sufficiently shared and stable **form of life**. In turn, the variety of **language games** increased and roughly five could be identified:

1. Asset management - the language understood by the original administration staff
2. IMPART - the new computer jargon understood by IMPART kingpins
3. Transformation talk - jargon introduced by the various sets of new rmanagers and consultants
4. Core/Satellite - investment strategy jargon *not* yet understood in the administration area and introduced by the Investment consultants
5. Consultant - project management jargon introduced by the Admin consultants

The asset management language game was the most widely spoken and understood, and it was the currency of the day-to-day operation within the administration function. Apart from a few IMPART kingpins who could understand both the IMPART language game and the asset management language game, most people were skilled only in their own particular language game.

My assessment is that 'acrimonious meeting' failed to establish a viable **commitment network** because of mishandling of the variety of **language games** and the variety of **forms of life**. Specifically, the consultant's language game dominated the planning initiative and effectively excluded all the other language games most notably the asset management language game. The unstable **form of life** hindered the **forming** movement in establishing a cohesive **commitment network**, and this necessitated stronger regulation through the **governing** movement. So it was difficult to see how a **commitment network** could be established through the consultant's project management methodology without embedding it in the asset management language game which was the only viable language capable of **governing** present commitments. In CPD terms, the **legitimizing** movement had *inappropriately* legitimized the consultant's language game as the **shared mode of discourse** for the Core/Satellite project. The general pre-conception was that the Admin Consultants were **experts** in project management, were being paid large fees to perform as experts in this capacity, and for this reason they felt pressurised to perform as experts which involved utilising their own specialised language. By failing to establish a viable **commitment network**, the 'acrimonious meeting' failed to perform and, via the **performing** movement, weakened the cohesiveness of the project **community**. Few issues were established as **shared public images** through the **normalizing movement** which inhibited the **forming** of **community**.

It can be seen that the **language** subsystem had generated within itself **loose complexity** which, via inter-subsystem movements, had spread throughout the whole project system. The Admin Consultants attempted to respond to this complexity by using a conversation type of **communication** as per the **conduit metaphor**. My assessment is that there was too much complexity within the **language** subsystem for this to be effective as the **taken for granted** worlds of the various sets of **experts** were too disjointed for a meaningful conversation that was attempting to bring about the **coordination** of **behaviour** and **expectations**. In other words, the **conversation design** of the project was not viable. The Admin Consultants were trying to **communicate** a plan rather than design a collaborative conversation whereby the administration area could build a workable plan as an 'act of shared creation'. Essentially, the Admin Consultants themselves had no **knowledge** with which to conceive a workable plan - the real **experts** in the situation were the longer serving permanent employees from within the company, not new recruits nor external consultants. The consultants were trying to act as **experts** who would deliver something rather than as facilitators attempting to utilise the expertise of incumbent **experts**.

7.2.3 Description of the intervention

The intervention started within the acrimonious meeting itself. Five key permanent staff members voiced their objections to the ineffectual plan. These people were:

1. The financial manager. He had also been a successful portfolio manager and had an excellent knowledge of the entire business.
2. The two IMPART kingpins. Even though they were protective of their IMPART knowledge and thus in a sense able to hold the entire administration area to ransom, they nevertheless had extensive knowledge of how the whole business functioned. They were also hard workers and keen to help.
3. The recently demoted administration manager. In spite of poor performance as a manager, he had an extensive knowledge of the business. He also had some idea of IMPART's functionality as he was managing the implementation of the package before the new management team was recruited.
4. The middle manager in charge of portfolio administration.

I observed the language and distinctions that these people used in voicing their objections. It became clear that they were angry at not being consulted, not so much through pride, but through sheer impracticality and wasted time. These five individuals each made differing observations about the plan, often agreeing with each other, and occasionally contradicting each other. In voicing their objections to the ineffectual plan, a cluster of common ideas seemed to emerge. In particular, they all seemed to be arguing for a plan that emphasized a 'minimum deliverable'.

Later that day, each of these people was informally interviewed. They were asked to give their opinions and ideas on how to plan a Core/Satellite implementation in the administration area. They were also asked to give a more detailed reaction to the ineffective plan and also to elaborate on what they meant by a 'minimum deliverable'. Their opinions were noted in rough as they talked. No interview took more than forty-five minutes. Each of the five individuals had by the end of the day given their private images. That evening, these images were clustered into common groupings so that overlaps between the private images could be combined into common images. Once the clusters were organised, a mode representation was designed and all clusters of images were incorporated into the mode of representation. Because there was no one to consult (it was now late at night), two representations were made in the hope that one at least would be meaningful to the five contributors. Appendices B1 and B2 shows the two representations. B1 emphasizes the minimum deliverable, and B2 emphasizes projects and milestones.

Next morning, these representations were individually reflected back to the five contributors. There were some small adjustments, but the general consensus was that both representations were useful, both accurately reflected each individual's contribution, that the plan was workable, and that they would support it. In the course of a hectic morning, an unplanned and impromptu meeting occurred

with the external I.T. consultants which had a good and a bad consequence. The good consequence was that they saw a workable plan which was supported by the administration experts. They immediately fed this news to the administration manager who of his own accord organised that the afternoon meeting discuss this new plan. This avoided political issues with the Admin Consultants who were sensitive from the acrimonious meeting of the previous day. The bad consequence was that the external I.T. consultants insisted on an additional step - the inclusion of a relational database to facilitate reporting of Core/Satellite portfolios. This proved later to be unnecessary, but the political nature of their involvement resulted in this additional requirement not being questioned until later on.

The afternoon meeting, which I will call the "harmonious meeting", attempted to 'turn' the conversation from communication into collaboration. The representations were photocopied onto clear overhead transparencies. The plan was not presented as a fait accompli. The plan was put forward as a starter plan to be manipulated and worked on by the whole management group during the session. It will be noticed that appendix B1's title ends with a question mark. Amendments to the plan were recorded by writing on the overhead transparencies with overhead pens. The overhead transparencies thus became shared space for the collaboration. During the meeting, the five contributors elaborated on the plan. As they did so and questions were raised, additions and corrections were made to the overhead transparencies as necessary. The corrected plan was then used as shared space to discuss project resource allocation. B3 is a photocopy of one of the overhead transparencies at the end of the meeting. It shows that most manipulations in shared space concerned resource allocation. There was little amendment to the original content of the plan as the real experts had already provided their contributions. By the end of the harmonious meeting, the administration management team had finalized an agreed plan for taking on the Core/Satellite investment strategy. Appendices B4, B5 and B6 detail the final plan.

At that stage the intervention can be considered to have been completed. The implementation of the plan was managed jointly by the Admin Consultants and the I.T. manager. Each person responsible for one of the identified projects submitted within a week a detailed plan of how they would tackle their task. This was done according to a set format in line with traditional project management ideas. B7 is a template that each manager of a sub-project was asked to complete. Thereafter, there were regular progress meetings. The intervention was a success in that it quickly alleviated a crisis. It effectively turned a poorly designed conversation based on ineffective communication into an effective collaboration in little over 24 hours. The results of the collaboration provided a workable plan to implement a mission critical project of unparalleled importance in the history of the asset management company. The Core/Satellite strategy was eventually implemented by the deadline date.

The following is a brief summary the intervention:

1. An inability to cope is becomes apparent.

2. Observe the design of the conversation which causes an inability to cope.
3. 'See' the language that experts use when they describe or talk about the problem situation.
4. Notice the distinctions which experts make.
5. Use these distinctions to design a mode of representation.
6. Synthesise and represent as best one can the private and shared images of the situation.
7. Reflect back the images via the designed mode of representation.
8. Amend the mode of representation if necessary *by manipulating the representation*.
9. Invite participants to correct content of the representation *by manipulating the representation*.
10. Participants are now collaborating - they are creating a shared object.
11. An ability to collaborate generates trust which reinforces collaboration.
12. Activities become easier to synchronize and co-ordinate.
13. Complexity diminishes.
14. People start performing against meaningful expectations.

7.2.4 Analysis of the intervention

1. The conversation type of **communication** as per the **conduit metaphor** was assessed as being inappropriate to deal with the **loose complexity** of the **language** subsystem.

2. **Shared meaning** needed to be established quickly via some *practical* form of **dialogue**.

3. **Experts** familiar with the **language game** of asset management were involved in a **dialogue**.

4. Table 1 and CPD provide insights to guide the design of the **dialogue**. The conversation should focus on **learning** and creating shared **knowledge**. It should tap into the **collective intelligence** of the **group** in order to surface shared **images**, **metaphors** and **assumptions**. The **taken for granted** background should be **unfolded and reflected upon** and **multiple representations** created so that **private images** could be expressed and considered by the **group**. These **private images** needed to be synthesised via the **worldmaking** movement into **shared public images** to enable the **forming** of **community** and the **designing** of **shared purpose**. **Humiliation and cruelty** would be avoided by allowing the experts to describe themselves and their own situation in their own way.

5. The **expressing** movement allowed the experts to generate **private images** through **viable conversation** by conducting informal interviews that used their *own familiar language*.

6. The **worldmaking** movement synthesised these **private images** into **shared public images** by clustering the common ideas into two **off-loaded representations**.

7. **Group learning** was achieved in the process of reflecting back to each expert the **shared public images** contained in the two **off-loaded representations**.

8. A 'minimum deliverable' emerged as a critical **shared public image** influencing the **designing** movement that established **shared purpose**. **Community** formed naturally around this **shared public image**.

9. The **transforming** movement influenced **language games** in the following ways:

- **shared purpose** was built from genuinely **shared public images** which emphasised a minimum deliverable in a generally familiar language that was *capable* of **governing** a **commitment network**
- **shared purpose** was **off-loaded** into representations that contained the structure of an actual plan as opposed to consultant language which was essentially hypothetical talk *about* plans and *not capable* of **governing** a **commitment network**
- **shared purpose** managed to reflect a more appropriate balance between the various **language games**

10. During the 'harmonius meeting' the experts who had been involved in the improvised **dialogue** formed **community** around their **shared public images** via the **forming** movement. Their cohesiveness enabled the **socializing** movement to exert a powerful influence on shaping **private images** of the other managers and consultants.

11. The **governing** movement strengthened the cohesiveness of the **community** as the rest of managers and consultants began to agree with and extend the expert's **shared purpose**.

12. The **metaphor** subsystem had enabled the **trusting** movement by creating a cohesive **community**, and enabled the **representing** movement by off-loading the **shared public images** as **manipulable representations**.

13. It was appropriate to conduct the 'harmonious meeting' according to the conversation type of **collaboration** for the following reasons:

- the plan to cope with Core/Satellite was not complete as the new managers and consultants had not had an opportunity to contribute their ideas (such as resource allocation)
- a **shared mode of discourse** had been established by the **representing** movement, and the **trusting** movement had created a climate conducive to **forming** a **collaboration network**
- **cohesive simplicity** needed to be established in order enable the **co-ordinating** movement and the **explicating** movement to limit the **loose complexity** of the **language** subsystem

14. Table 1 shows that all the conditions for a conversation type of **collaboration** had been achieved. The **shared mode of discourse** was generated from two **manipulable representations** that represented the expert's images and purpose. These representations were now **off-loaded** onto the **environment** where they could be used as **manipulable representations** suitable to act as **shared space** in a collaboration. Thus a shared **interpretative system** had evolved and provided a **basis for viable, understandable and meaningful conversations**.

15. The experts' plan was extended via the **designing** movement until there was agreement that the modified plan could act as both an appropriate **synchronization mechanism** for **co-ordinating** Core/Satellite **activity**, and also as a regulator suitable for **governing** an on-going **collaboration.**

16. After the 'harmonious meeting', the **explicating** movement established detailed commitments by requiring each individual responsible for a sub-project to complete the project forms (see appendix B7).

7.3 PROJECT 3 - RELATIONAL DATABASE

7.3.1 Background

This project arose from the planning process described in the previous section and took place in April 1997. It can be seen from appendix B4 that the plan asks for a "relational database similar to Fred's system" in order to allow "flexible reporting" and also to "analyse and rationalise" reporting. It can also be seen from appendix B5 that two sub-projects "analyse reports & spreadsheets" and "reporting database" are required to "consolidate a reporting capability". One of the major concerns in implementing the new investment strategy was to convince external clients, which included the parent life assurance company, that Core/Satellite was a sound strategy which would dramatically improve investment performance. This involved an educational and selling process conducted mainly by Marketing (4th Floor). However, the clients that Marketing were talking to about Core/Satellite were also concerned about the reports which they would be receiving under the new strategy.

Each month clients would receive a series of reports detailing the status of their investments. These reports were often inaccurate. The whole process of producing the reports included a long chain of poorly managed events and procedures which no one person or department was able to describe completely. The most serious inaccuracies were caused through incorrect calculation of unit prices. For a unit price to be correct, a whole month's worth of trading had to be correctly processed by IMPART. The process of obtaining a unit price calculation from IMPART was locked up in the head of an IMPART kingpin. There were many problems in obtaining accurate unit prices and therefore accurate reports. The whole month-end process was a chaotic scramble that routinely necessitated the IMPART kingpins to work 36 hours without sleep. Serious mistakes would require the whole

month-end unit pricing process to be rerun. Late unit prices caused as many problems as inaccurate unit prices.

It was difficult, therefore, for Marketing to sell the concept of Core/Satellite without getting involved in conversations about reporting. Clients, already concerned about poor service, wanted to know the implications, especially what kind of reports they would be getting, and how accurate they would be. At the time that Core/Satellite was being 'sold' to clients, no discussions had taken place between Marketing and Administration about reporting. In particular, Portfolio Admin, whose main purpose was to compile accurate reports and statements for clients, had no idea of the implications of Core/Satellite on their reporting function. Basically, the details of Core/Satellite had not been completely thought through by the time it was sold.

The relational database sub-project came about as a result of an unfortunate sequence of events. The external I.T. consultants, who had been flown in by the managing director at great expense from overseas, had suggested the inclusion of the relational database as part of the plan. In effect, this was their sole contribution to the proceedings. After they had reassured the managing director, they then departed. It will be noted from appendix B5 that the very first task is to "Understand Core/Satellite". If this concept had been fully understood at the time that the planning process was taking place, it would have been understood that "Fred's system" mentioned in B5 could, with a few changes, already cope with the client reporting requirements. When this became apparent a week or so later, "Fred's system" was immediately changed, and the relational database was no longer critical to the Core/Satellite implementation.

It was decided to continue with the relational database project. Politics played a part in this, because by this stage the relational database had taken hold in management's mind. The I.T. manager also felt that a relational database would be an important part of the longer term I.T. strategy. There was still a short-term need to analyse and rationalize reports, and a skilled data analyst could develop a data model of the asset management operation as a by-product of this process. In the longer term, the I.T. manager felt that reporting should be done from a flexible relational database, but not as part of the minimum deliverable for Core/Satellite, and "Fred's system" should be replaced for a number of other reasons not pertinent to this discussion.

Section 2.2 outlines a point count method for assessing the potential for success of a software project, and this will help to show why the sequence of events by which the relational database project was set up limited its chances of success:

User Involvement:
There was no user involvement at all. The project originated totally from I.T. sources. Firstly, external I.T. consultants deemed it to be necessary during a three-day visit. The project could have been

abandoned when it became apparent that it was not needed, but the I.T. manager then required that it continue for other reasons. It was explicitly agreed that the relational database project should not hinder progress toward Core/Satellite implementation. Since the whole organisation was already over-stressed, this effectively meant that it was difficult to talk to anyone about the relational database. (Score: 0 out of 19)

Executive management support:

In word, executive management supported the relational database, but in deed, because they had no conception of what exactly the relational database was going to deliver, they *did* nothing to support the project. There is a great deal of difference between *saying* that one supports something and *acting* to show that support. Executive management definitely did *not* support the I.T. manager's ideas of why the relational database was important in the longer term. (Score: 0 out of 16)

Clear statement of requirements:

There were no user requirements at all. Users were relatively uninvolved in the project. Marketing had never approached Administration or I.T. to specify new reporting requirements as a result of their understanding of Core/Satellite or discussions about reporting with clients. Administration themselves, particularly Portfolio Admin, could not specify requirements because they did not understand what Core/Satellite meant. (Score: 0 out of 15)

Proper planning:

The essence of the plan was as follows: while analysing and rationalising reporting, try to develop a data model of the organisation; disturb as few people as possible because nothing should interfere with implementation of Core/Satellite; check the final model to see the degree to which it can support the Core/Satellite reporting requirements of Marketing, Portfolio Management (i.e. asset management), Portfolio Administration and I.T. (especially IMPART). Although there are problems to this plan which will be discussed, it is not totally unworkable. A good data analyst, with limited explanation from users, can often derive a good data model from an organisation's reports and spreadsheets. I will score this generously. (Score: 11 out of 11)

Realistic expectations:

Only the I.T. manager had realistic expectations of the project. Once "Fred's system" had been amended to deliver a reporting capability, it was difficult for anyone else to understand why we were continuing with the project. Executive management had never had realistic expectations to begin with. (Score 0 out of 10)

Smaller project milestones:

The relational database project was in itself not a big or complex project for a competent data analyst. The programming needed to implement it was also fairly easy. But, because it essentially a by-

product of another project, it did not have its own carefully planned milestones. For example, because of the pressure on the programmers it was never clear when the relational database would be implemented. (Score 0 out of 9)

Competent staff:
There was no problem with the technical competence of the staff. (Score 8 out of 8)

Ownership:
If there was any ownership, then it was by the I.T. manager. But the administration manager, as the executive sponsor, was focused on short term delivery and never actually agreed on the longer term importance of a relational database. (Score 0 out of 6)

Clear vision & objectives:
The vision and objectives were unclear and not shared. (Score 0 out of 3)

Hard-working, focused staff:
Everybody was working hard, but this effort was not focused on the relational database. (Score 0 out of 3)

It can be seen from this that the project was set up with a less than 20% chance of success. This is an example of the sort of software project which regularly fails in commercial organisations. Although a relational database was installed about six months later based on the data model developed during the project, this does not detract from the fact that the project was ill-conceived. In the short term, it did not matter a great deal whether the relational database succeeded or failed. However, an employee or consultant operating in a large commercial organisation is often placed in awkward, seemingly nonsensical situations, has to make sense out of them, and ultimately perform according to expectations. In attempting to make sense out of the relational database project, a much larger problem was encountered that could not be ignored as it endangered the whole implementation of Core/Satellite.

It became apparent after a few week's work on the reporting and relational database projects that the organisation as a whole had no consistent understanding of what the words "Core/Satellite" actually meant. It was fairly easy to develop a relational data model as a by-product of "analysing reports and spreadsheets". Appendix C1 represents the high level data model in an entity relationship diagram (ERD). This model had to support new Core/Satellite client reporting and therefore had to be verified by various internal experts. It was during this verification process that it became clear that there were a number of interpretations of what Core/Satellite meant.

7.3.2 Framing the intervention

The situation was framed as being a problem of conversation design. There were four conversation design problems associated with the relational database project:

1. It was not conceived as a collaborative project - it was not an act of shared creation - and lacked user involvement necessary for success.

2. The language game of the relational database project centred on a private method, namely data modelling via ERDs. In other words, how was it possible to verify a model when the people being asked to verify the model have no skill in using the modelling language? There was no shared mode of discourse.

3. The project was a small part of a larger nonviable conversation. A new language game, Core/Satellite, was starting to be spoken but its meaning was unclear, and people were generally not skilled in this language game. At times it seemed as if the words "Core/Satellite" were just the label of an event, a convenient title for a whole range of frantic activity. Even more worrying was that there was no expert to provide a final meaning. The Investment Consultants, who had introduced the term, were certain about the overall principles, but had tailored their ideas to the practicalities of the organisation. For example, certain pre-existing agreements, contracts and mandates from clients constrained the form of the Core/Satellite strategy. Thus, the Investment Consultants were learning the meaning of the words from those they were supposedly educating. While they were still in this learning process, Marketing were already selling the concept to customers. Most managers had a good idea of the principles, some employees had a vague idea, but no one had the final answer as to what it *exactly it meant in practice*.

4. Allied to 3 above, there was a great deal of conversation about what Core/Satellite meant. But the conversation type was communication, not dialogue or collaboration. There was no attempt to design a conversation where the private understandings of the words "Core/Satellite" could be synthesized into a shared public image as an act of shared creation. In other words, it is futile to attempt to *communicate* meaning when there is no shared system of interpretation to give that communication meaning. A different form of conversation was needed - one that could build a shared system of interpretation.

The four points above can be rephrased explicitly in terms of CPD. The overall **form of life** was being transformed. Admininstration, Investments, Marketing, I.T. and other forms of life were all involved in the process **forming** a new **commitment network**. The **shared purpose** of the whole asset management company was to transform the **language game** through the **transforming** movement so

that the Core/Satellite **language game** would become capable of **governing** a transformed, viable and cohesive **commitment network**.

The relational database formed part of this **commitment network** even though it was a poorly conceived project. My role corresponded to a standard **form of life**, namely computer systems analysis, and data modelling would normally be used as the standard **language game** to govern my activity. Through the **legitimizing** movement, ERD is *inappropriately* standardized by many I.T. methodologies as a **shared mode of discourse**. My experience is that it seldom serves as a **basis for viable, understandable and meaningful conversations** because it is a **private method** that is seldom meaningful outside (and often within) I.T. circles. The intervention would have to design its own **shared space** in order to enable **viable conversation**.

The essential task of this intervention was to create a data model of the asset management company's operation as it would function under the Core/Satellite strategy. Essentially, I had to model the Core/Satellite **language game**. In order to do this, I would have to collaborate to some extent with all the various **forms of life** that were busy **forming** the new **commitment network**. I suspected (rightly) that there was much misunderstanding and confusion concerning the meaning of the terms Core/Satellite. I also anticipated little interest, commitment and user involvement. However, if I re-framed my purpose as that of helping others to understand Core/Satellite, then it might be possible to establish a **shared purpose** and assist in **forming** some kind of **community** around the relational database project. Developing the data model could be accomplished through a conversation type of **communication** with a conscious attempt at 'remaking' the worlds of others and their **forms of life** as per the **world re-maker** metaphor. I could ask certain questions and individuals could communicate their answers. This would entail trying to understand the worlds of others. I would represent these answers using a **private method**, namely ERD, but would not be able to use this standard **mode of representation** to verify that the content of the ERD actually modelled the Core/Satellite language game accurately. I would need to devise a **re-representation** of the ERD that would not only allow the data model to be verified, but enable **learning** so that the **shared meaning** of Core/Satellite would crystalize in the **collective intelligence** of the user **group** to the extent that they would actually be in a position to verify the data model. In other words, creating the ERD could be achieved through the conversation type of **communication**, but verification of the data model would require **dialogue** and **collaboration**.

7.3.3 Description of the intervention

In order to 'turn' the conversation, it was necessary to design a mode of discourse whereby it would be possible to have viable conversations with various involved parties to determine firstly the meaning of Core/Satellite and secondly whether the data model supported this meaning. Experience

has shown that while ERD is an effective method for modelling data it cannot generally be used as a shared mode of discourse for conversing with non-systems people. The ERD was retained as a useful private method, but was re-represented for purposes of conversing with users. The asset management company was proficient in the use of spreadsheets, and it seemed that this would be an appropriate metaphor upon which to design a more understandable mode of representation. The concept of a relational database is not always an easy concept to grasp. By metaphorically redescribing it as a series of inter-linked spreadsheets, it is possible to converse in a metaphor with which everyone is already familiar. Appendix C2 re-represents C1 and also simulates four months of data representing my best understanding of the Core/Satellite concept. The icon in the top left corner represents a puzzled face and is an invitation to correct either the model or its data. Once again, the strategy is to represent something in familiar language, stress that it is 'a best attempt' and that it may need correction, and ensure that any corrections are done by manipulating the representation not merely by verbal communication.

In order to assess the suitability of the mode of representation, I showed the representation in C2 to a few people who understood the business and had some grasp of the principles of Core/Satellite. I learned that it was extremely difficult for most people to work through unassisted. So I developed a routine of walking people through the model in a set sequence that involved colour coding. This was more helpful and could usually be accomplished in an hour. It could be done with individuals or groups. Some individuals would then take the representation and work through it in their own time. (The one included as a sample, C2, shows in the bottom left hand corner that someone has spotted some errors.) The advantage of leading people through the model was that they were more able to concentrate on whether the data structures were correct.

Having established that this was a viable mode of discourse, I was encouraged by the I.T. manager to test it as widely as I could throughout the company, especially those area that were associated with reporting, and in particular client reporting. At first I was disappointed with the outcome. It seemed that either the representation was *too* correct or that it was not understandable or that nobody was really interested in the process. Two things encouraged me to continue. The two IMPART kingpins, who together had broad detailed knowledge of the whole business and unique knowledge of the systems aspects, paid great attention to the representation. They disagreed with some of the content, argued about a crucial aspect of the data structure, and when these had been corrected and resolved, they then said that they felt happy that IMPART would be able to handle Core/Satellite (as they understood it) and that the data model would be able to be populated from IMPART. In other words, two experts contributed to the representation of Core/Satellite. This gave me more confidence, and I started to think that the lack of response was caused partly by the model being basically correct.

The second piece of encouragement was sparked by the longest serving employee in the administration area who I will call the "Senior expert". In particular, she had a wealth of historical

knowledge, especially from prior years when the company was performing well. At the end of the walk-through she expressed a concern which went something like this:

SENIOR EXPERT: Is ABC Retailers part of Core/Satellite?

ANALYST: Yes

SENIOR EXPERT: Well then, something is wrong. I can not spot any mistakes in the logic or the simulated data, but the column headed TotalUnitHoldingD in the Unit Holding spreadsheet holds *fractions* of a unit. Look, the first figure you have there is 1562.5. Is this correct? Is it intentional that you have included data that can hold .5 of a unit?

ANALYST: Yes. The way Core/Satellite seems to work is that each company effectively holds units in a combination of Core or Satellite portfolios depending on the mandate that they choose and that the unit holdings will seldom work out to be whole numbers.

SENIOR EXPERT: Well, something is wrong. ABC Retailers will never ever agree to be part of such a strategy. They have emphasised over and over that they want their portfolio always to be separately managed as an independent entity. How do you plan to report partial unit holdings? Would .5 of a unit holding be reflected on a client's statement?

ANALYST: I understood that they would see only the monetary equivalent of the unit holding. The adjacent column holds the market value of R1,562.50.

SENIOR EXPERT: ABC Retailers check their statement to the last cent. They keep their own records and compare them with our statement. They want to see how we work out that amount, so they will have to know about the fraction of a unit. If this representation is correct, the implication is that ABC Retailers have agreed to be part of a unitized strategy, but I know they will never agree to this. You'd better check what Marketing has told them about Core/Satellite. Either Marketing or ABC Retailers has misunderstood the concept of Core/Satellite.

This excerpt shows that the mode of discourse allows viable conversation. Even though the Senior Expert is no expert on Core/Satellite, she has been able to identify something that does not seem meaningful. By attending to this, and working through it via the shared mode of discourse, meaning can be gradually built. In fact, this exchange had serious repercussions. Marketing were equally worried when they saw fractions of a unit in the representation, but agreed also that the logic of the representation seemed correct. Essentially, something had gone wrong during the marketing of Core/Satellite. Top management were required to make decisions as to how to rescue the situation. I was not party to the final discussion, but interim discussions revolved around what to show on client reports. These were the kind of questions being asked: Do we round units to the nearest whole number? Do we just show monetary amounts? Do we show full workings? Maybe now is the time to get clients to accept different types of reports? Are we being totally honest with clients about the concept of Core/Satellite? Do clients really understand what we are selling them?

The final outcome was that Marketing had to revisit their clients and in effect 'come clean' about their original marketing story. The net result is that Core/Satellite lost about a quarter of its portfolios which

then became separately managed portfolios (SMP's). Not only did this reduce the impact of the Core/Satellite strategy, but it embarrassed the organisation in the midst of what was supposed to be its big transformation.

Effectively, this marked the end of the relational database project. No other serious objections were made about the data model or the re-representation of it. The project was put on the shelf for a few months and then implemented by the programming team. It was never anticipated that the intervention would take such a dramatic turn. But by the time the misunderstandings between the investment and administration functions had been resolved, there was a new understanding of the meaning of the words "Core/Satellite" and the implications for client reports had been satisfactorily sorted out. The intervention never succeeded in 'turning' the conversation so that it became a truly collaborative project supported by committed user involvement. But it did turn the conversation enough to provide solutions to the four conversation design problems stated at the beginning of the previous subsection. In spite of a low potential for success, some sense was made of the project by redesigning its critical conversations. I was able to 'look through' Wittgenstein's concepts of activity, forms of life and language games and 'see' the project clearly. I 'saw' that management were concentrating on activity and deliverables, but had neglected to manage the language game governing these activities. I 'saw' that by failing to establish the *meaning* of two words they had jeopardised their whole transformation effort.

The following is a brief summary of the intervention:

1. Forms of life involved: asset management, asset administration, marketing, systems analysis.
2. Language games:
- The existing asset management language game - the widely spoken, public language.
- Core/Satellite - this was the new public language with unclear meaning.
- IMPART - a private systems language of which only two people knew the whole vocabulary.
- Entity relationship diagramming (ERD) - a private method used by systems analysts to model data.

3. Metaphoric redescription of ERD and relational databases as inter-linked spreadsheets.
4. Design a mode of representation to re-represent the ERD using the spreadsheet metaphor.
5. Add meaning to the representation by simulating Core/Satellite data.
6. Establish whether the mode of representation is generally meaningful.
7. Use the Core/Satellite representation to design a shared mode of discourse that allows viable conversation about both the meaning of the words Core/Satellite and also the validity of the data model.
8. Promote a conversation spanning all forms of life which uses the shared mode of discourse to establish the meaning of the words 'Core/Satellite'.

7.3.4 Analysis of the intervention

This subsection will outline those aspects of the intervention that have not yet been described explicitly in CPD terms. The ERD, a **private method**, represented the **private images** of the systems analyst and various users who had communicated to the analyst about Core/Satellite. These images would remain private until a **worldmaking** movement could be enabled that would allow the **private images** contained in the ERD to be shared in a meaningful way and become **shared public images**. Through **metaphoric redescription** of the ERD as a series interlinked spreadsheets, a new **mode of representation** was conceived. With the aid of a facilitation technique to walk people through the re-represented ERD, the new representation allowed people to understand each other's **private images** in terms of a familiar **metaphor**, namely spreadsheets. This form of **worldmaking** enabled **learning** about Core/Satellite and a **shared public image** began to develop. **Learning**, the creation of shared **knowledge,** unfolding background and making it **present-to-hand**, creating **shared meaning** from **loose complexity** are goals of the conversation type of **dialogue** and the **metaphor** subsystem. Although this conversation was far removed from a prototypical **dialogue** as described in subsection 3.3.3, the conversation did achieve many of these goals and in doing so created a bridge to the **conversation** subsystem.

The **re-representation** of the ERD enabled the **representing** movement to counter the **legitimizing** movement which inappropriately standardizes ERD as a **shared mode of discourse** for data modelling. The **re-representation** had created a **designed, reusable mind tool** that could be used as **shared space** in a series of conversations. **Collective intelligence** of the **group** had now been **off-loaded** onto the **environment** in a tangible form that consolidated understanding of the meaning of Core/Satellite, and made it explicit (**explicated as foreground**). This process helps to create **cohesive simplicity** out of **shared meaning**. The **shared mode of discourse** enabled a **synchronisation mechanism** that allowed a series conversations concerning the meaning of Core/Satellite to be linked together in a larger, synchronised conversation. In this way, a **collaboration network** came about through the **forming** movement and the **governing** movement.

For example, Portfolio Admin (esp. observations of the Senior Expert) became synchronised with a separate conversation involving Marketing. Both parties focused on the same **shared space**, both were able to identify a shared problem, and together they were able to explicate this shared problem to a third party, the management group, using the **shared mode of discourse**. This example illustrates that the **conversation** subsystem had generated a sufficient degree of **cohesive simplicity** to enable three important movements:

1. The **explicating** movement was able to comprehend the nature of Core/Satellite commitments with sufficient precision to identify a critical problem concerning the **commitment network**. It was able to identify that commitments to certain clients could not be honoured. It had identified that **loose complexity** existed in the **commitment network** through insufficient understanding of the Core/Satellite **language game**. However, now that this had been identified, the **shared meaning** generated by the **metaphor** subsystem would begin to counteract this problem via the **transforming** movement.

2. The **socializing** movement enabled a series of **viable conversations** by enabling them all to re-use the same **designed, reusable mind tool**.

3. The **expressing** movement was enabled by increasingly **viable conversations** which allowed individuals a viable way of **expressing** their **private images** and an efficient method of **learning** about Core/Satellite.

7.4 PROJECT 4 - ADMIN TRANSFORMATION

7.4.1 Background

The main reason why the Admin Consultants had been hired was to transform the administration function of the asset management company. The Core/Satellite initiative had temporarily sidelined the Admin transformation project. By June 1997, Core/Satellite was in operation and it was possible to turn attention to Admin transformation once more. The consultants' initial efforts at planning the Core/Satellite implementation had caused hostility from the administration staff. In order to regain confidence and to avoid further acrimonious meetings, the consultants requested that they play a background role during the initial planning of the transformation. They would take responsibility for managing the implementation of the solution but not for conceiving it. The consultants and Admin management suggested we repeat a similar process to that which had been used in Project 2 in dealing with the Core/Satellite planning crisis.

The transformation was necessary because inefficiencies and inaccuracies had grown to an unacceptable level. A new management team had been brought in to rescue the company, and a total transformation of the administration function was expected. The company was losing clients.

7.4.2 Framing the intervention

This project was concerned with transformation. In terms of CPD, it required the **language** subsystem to be influenced through two critical movements:

1. The **transforming** movement would have to transform the current Admin **language game** and bring about a new language game capable of **governing** a new **commitment network**. This would entail synthesising **shared public images** from **private images** through a **worldmaking** movement and then creating **a shared purpose** through the **designing** movement. This would be achieved through a conversation type of **dialogue** focused on unfolding and synthesising **multiple representations** of the administration function's future through a **group learning** process.

2. The **explicating** movement would attempt to create a clear unambiguous set of commitments in order to facilitate the **forming** of a cohesive **commitment network**. This would only be achieved if the **conversation** system could convert the **shared meaning** established by the **metaphor** subsystem into **cohesive simplicity** through a conversation type of **collaboration**. Thus it would be necessary through the **representing** movement to establish a **shared mode of discourse** suitable for **forming** a **collaboration network**.

The **transforming** movement and the **explicating** movement would have to be mutually reinforcing in order to create a new cohesive **commitment network**.

7.4.2 Description of the intervention

Unlike the previous projects where a poorly designed conversation existed and had to be 'turned' into a collaboration, this intervention needed to design a new, viable collaboration from the beginning. This was done in a number of steps.

Step 1 - Objective and Principles

The top administration managers agreed on a set of transformation principles as a group. This involved five interviews with each manager which lasted no more than 45 minutes. The results of these interviews were consolidated in a rough draft which was discussed by the top management team as a group. During this meeting they agreed on a concise objective and affirmed a joint set of principles (see appendix D1).

Step 2 - Private Images

The top managers were mostly new recruits, whereas the middle managers were longer serving employees with hands-on, detailed experience of the operation. For this reason, the middle managers were asked to devise the transformation plan, but guided by top management's objective and set of principles. A second round of twelve interviews was conducted. This included the five top managers and seven middle managers, and covered areas such as portfolio administration, data capture, I.T. and IMPART, and accounting. They were given time beforehand to digest the objective and principles. They were also told that each interview would be asking for their ideas on the following:

1. What is their reaction to the objective and guiding principles?
2. What is wrong in the administration function?
3. What should be done to transform the administration function?
4. Do they have any ideas on a suitable structure for a transformed administration function?

Rough notes were kept, but anonymity was assured and people spoke freely. A precedent of anonymity had been set in Project 2. Some interviews took as long as two hours and were similar to counselling sessions; people used the interviews to work things out in their own minds and sometimes asked for their ideas to be reflected back to them so that they could re-digest them and also make sure that what they had said was correctly interpreted. Sometimes a new idea would crop up and talking about it interactively would help to clarify it.

Step 3 - Shared Public Images

After the interviews were completed, I clustered the ideas and comments into groupings and derived an appropriate title for each cluster (see appendix D2). In this way, each idea was still visible within a cluster, but it was synthesized into a shared whole represented by the title of the cluster. Participants could see their own ideas, see how they corresponded to other people's ideas, and yet have their ideas remain anonymous. It was not practical to synthesize the twelve ideas on structure so these were merely summarized (see appendix D3). Because of the length of these documents, I designed a method of representing them on a single page (see appendix D4). This summarized the objective and issues of concern for the transformation project. This whole package, D1, D2, D3 and D4, was fed back to the twelve managers as a group. They agreed that the objective, principles and issues should be used as a basis for re-designing the administration function.

Step 4 - Designing the collaboration

The next step was to design a shared mode of discourse to enable viable collaborative conversation which was necessary for the process of designing a new approach to administration. I had used process dependency diagrams (PDDs) effectively in the past in these sorts of situations. Although they are private methods of systems analysis, they are much more acceptable as a public language than, for example, ERDs. Firstly, it is easier to relate to processes with which one is involved every day than to abstract data structures which are never normally thought about. Secondly, the notation of PDDs is simple and fairly intuitive. Whereas one needs training to understand an ERD, people usually make sense of PDDs without any prior training. If training is needed, it can usually be accomplished within 30 minutes. *Drawing* meaningful PDDs *is* an art gained with experience, but *reading* a PDD need not be problematic.

Because I was uncertain of the suitability of PDD as shared space for the Admin Transformation, I needed to test it out beforehand. I represented as best I could the administration process as a PDD. I

validated this with two of the middle managers who would be involved in the collaboration workshop. I observed that they had no problem relating to the mode of representation, and they were able to point out some mistakes with the content. In these cases we corrected the mistakes by manipulating the diagram. Lines were crossed out, a new box was added and so on. Appendix D5 shows the corrected PDD. From this it was decided that it would be acceptable to use the PDD as a public method for the collaboration.

Step 5 - The Collaborative workshop

The seven middle managers and two consultants took part in a two day workshop. The purpose was to design a strategy whereby the administration area could achieve its objective of "Accurate daily unit prices for all portfolios". The participants had a set of guiding principles (D1), a comprehensive set of ideas clustered into issues (D2), 12 sets of ideas about structure (D3), and a one page representation of the transformation project (D4). These had all been pre-digested, shared and agreed. They also had a manipulable shared space where they could represent their solutions (D5).

On the first day, the group split into two sub-groups. Each sub-group designed a first cut solution and presented it back to the full group at the end of the day. The next morning they chose one option, but re-represented it using ideas from the discarded option. With a few notable exceptions, the group ended up sharing similar ideas about what the solution should look like. They decided on what they called a "Hub/Team" strategy, and represented it with their own modified version of a PDD (see appendix D6). It can be seen that they extended the normal notation of PDDs to include their own notation. They also modified the original PDD (appendix D5) and then overlaid the Hub/Team strategy. In this way, they not only modified the original representation, but they also modified the original *mode* of representation to make it more meaningful and useful.

Step 6 - Approval and implementation

The consultants took over from this point as they were responsible for implementation. The Hub/Team strategy was presented to the five managers for their education and approval. In order to make it more understandable, the Hub/Team approach was re-represented for the sake of simplicity. Appendix D7 shows the Admin consultant's re-representation of the Hub/Team strategy, and Appendix D8 shows the responsibilities of the Hub and the Portfolio team respectively. The five managers agreed that the solution abided by the objective and principles, and that it addressed the issues unfolded prior to the workshop. The Admin consultants then organised and managed the implementation.

The following is a summarised description of the intervention:

1. Top managers express private images about objectives and guiding principles.
2. Objectives and principles are synthesized into a shared image and shared purpose.

3. All managers express private images about transformation issues.
4. Transformation issues are synthesized into shared public issues.
5. A private language (PDD) is tested to see whether it can function as a public language.
6. PDD is chosen as an acceptable mode of representation and a shared mode of discourse.
7. A representation is made of the administration process via PDD.
8. The final design of the collaboration includes:
- four pre-agreed, shared filters - D1, D2, D3, D4
- a shared manipulable space - D5
- a forum for conversation - a two day collaborative workshop.
9. The collaborators amend the mode of representation to suit their purposes .
10. The collaborators manipulate the representation in shared space.
11. A new language game, Hub/Team, is designed which is aligned to the original purpose.
12. Top management agree to learn and speak this new language.
13. End of intervention - consultants manage the implementation of the new language game.

7.4.4 Analysis of the intervention

The **worldmaking** movement involved one-on-one interviews allowing the **expressing** of **private images** and a clustering process which grouped these images and issues into common topics. At this stage the **private images** were not yet ready to be synthesised into **shared public images** because of a number of conflicts between the private images. The main emphasis was surfacing **assumptions** and **images**, **learning** about each others issues, generating **knowledge**, and unfolding **multiple representations** about Admin's problems and its future. The **designing** movement was beginning to influence **shared purpose** in the direction of 'Accurate daily unit prices for all portfolios'.

The systems analysis **form of life**, through the **legitimizing** movement, promotes a variety of process modelling techniques as **shared modes of discourse**. PDD, although it is a **private method** associated with I.T. systems analysis, is often appropriate as a **shared modes of discourse** as long as it is introduced with care. PDD was tested to see if it could function as a **shared mode of discourse** by using it to represent the current Admin processes. It proved to be an acceptable **mode of representation** and enabled a **viable conversation** that allowed a **shared public image** of the current Admin process to be represented. This **representing** movement provided a convenient bridge between the **metaphor** subsystem and the **conversation** subsystem because it was a **manipulable representation**. It represented the current process as a starter model that was then manipulated as a **collaboration** in **shared space** toward the new Admin process. The **representing** movement and the **expressing** movement began to reinforce each other. The representation provided an acceptable **shared mode of discourse** which enabled the **forming** of an effective **collaboration network** thereby enabling **viable conversation**. The **expressing** movement focused on incorporating new

private images into the evolving representation. In this way **private images** became synthesised into the new PDD process model as a **shared public image**. The 'hub/team' **metaphor**, which underpinned this **shared public image**, and the objective of 'Accurate daily unit prices for all portfolios' were merged in the **designing** movement to create a new **shared purpose**. Previous disputes between **private images** were resolved to the extent that a cohesive **community** was able to **form** and develop **trusting** relationships that strengthened the **collaboration network**.

The mutually reinforcing influences of the **representing**, **expressing** and **trusting** movements enabled the **metaphor** subsystem and the **conversation** subsystem to influence the **language** subsystem in a mutually reinforcing manner through the **transforming** and **explicating** movements. The transformation of Admin was successfully conceived during this intervention. The project management of the **activities** needed to bring about the new Admin **form of life** was handled by the Admin Consultants who designed a way to use the new PDD as a **synchronisation mechanism** to co-ordinate the on-going project.

7.5 PROJECT 5 - NEW DEALING SYSTEM

7.5.1 Background

This project was a critical part of the implementation of the previous project, the Admin Transformation project. Many data inaccuracies were caused by the fact that the dealing system, which traders used to buy and sell assets in the stock market, was not integrated with IMPART, the administration system. So every day each trade had to be manually captured into IMPART from printed reports produced by the dealing system. This operation was resource-intensive, error-prone and caused time delays. The intention was to provide a direct interface between a new dealing system and IMPART, so that the manual data capture of trade transactions could be eliminated.

A large number of people were affected by the project:

1. Portfolio managers used the system to place orders with the dealers and to receive confirmations about trades. They also used it to keep track of their portfolio asset holdings.
2. Dealers used the system to execute the orders and trade in the market.
3. Data Capture received reports of trades and recorded them on IMPART.
4. I.T. developed and maintained the existing system which I will call the "VB system".
5. The compliance officer needed to monitor that trading was legal.

There were a number of issues:

1. The VB system was user-friendly and much liked by the portfolio managers and the equity dealers. The dealing manager had specified the system requirements; the system was programmed in-house and he was proud of it. However, it had no functionality to support bonds, futures and other asset types. So the dealers of these latter instruments welcomed a new system. Also, although it was very friendly, the VB system was not completely robust. There was only one person, whom I will call the "VB programmer", who could maintain it. Management wanted to reduce its dependency on a somewhat erratic system and a single programmer. The VB programmer supported this move as he was tired of maintaining the VB system; the system had never been designed as the long-term solution, and he felt it should be replaced.

2. The pain of the project would be felt by the dealers and portfolio managers, but the benefits would be felt in Administration, especially Data Capture. Obviously, the whole company would benefit from improved, accurate service to the clients. Also, portfolio managers would receive more accurate and up-to-date information about the status of their portfolios. But, there was a good deal of animosity between administration which included Data Capture (3rd floor), and portfolio management and dealing (4th floor). In particular, the administration manager and the investment manager were on particularly bad terms. The politics of trying to get the 4th floor to assist the 3rd floor, which could quite possibly result in extra work for the former, was a problematic issue.

3. Speed of implementation was critical to the Admin transformation project as a whole. A new dealing system was seen to be the biggest sub-project and the one most likely to improve service. Software package selection or in-house development of a new system would take at least six months to implement. A quick solution was needed.

4. The whole organisation was under strain. Scheduling time with people would be difficult. It would be difficult to get groups of people together at short notice. Also, because there were so many daily crises, the dealing project would have to work in and around these crises.

The original reaction was that we should evaluate and select a suitable software package. But this was seen as a very time-consuming exercise. Another option was to adapt the VB system to interface with IMPART. This was the quickest and cheapest option, and the programming involved would be quite simple. The VB system could also be amended quite quickly to handle bonds. But management were not happy about being dependent on a single VB programmer. However, IMPART had dealing and ordering modules. Even if a full package evaluation was conducted, these modules would be very likely to end up on a short-list because they were certain to be compatible with the IMPART administration modules. Also, one of the portfolio managers had previously worked in a company which used IMPART dealing and ordering, and found them satisfactory. So, for the sake of speed, a decision was made to install the IMPART modules, test them and use them. If however during the test

some major problem was found, an adapted VB system could be used as a short term interim solution.

7.5.2 Framing the intervention

Through the **designing** movement of the previous project, the Admin transformation project, an initial **shared purpose** had already been established for this intervention; the **forms of life** of ordering, dealing, data capture and I.T. required the **forming** of a new **commitment network** under the **governing** influence of a new **language game** that emphasised data accuracy through IMPART software. A **dialogue** would be needed to unfold **shared meaning** and influence the creation of the new **language game** through the **transforming** movement. The **collaboration network** would be large with a significant amount of pre-existing distrust which would need to be countered by **forming** a more cohesive project **community**. This would influence the **forming** of better relationships within the **collaboration network** through the **trusting** movement.

The I.T. **form of life** is increasingly **legitimizing** prototyping as **a shared mode of discourse** for software implementation. A combination of IMPART modules and a Model Office environment would provide a manipulable **shared space** appropriate for a prototyping approach. A **shared mode of discourse** would be enabled by this **shared space**. The **representing** movement would allow **shared public images** to be represented in the **shared space** by manipulating the settings and operation of IMPART in a test environment. In other words, IMPART and the Model Office provided an **environment** onto which **shared public images** could be **off-loaded** and tested. It also provided a common, viable medium for the **expressing** of new **private images** needed by the conversation type of **dialogue**. A form of **dialogue** would be necessary to generate shared **knowledge** about IMPART and business processes so that the design of a new ordering, dealing and data capture process could be synthesised into a **shared public image** and **shared purpose**.

The Model Office would provide a forum for a conversation type of **collaboration**, but in order to achieve **viable conversation**, it would be necessary through the **designing** movement to develop a **synchronization mechanism** capable of **governing** the sizeable **collaboration network** and capable of **co-ordinating** the project's **activities**. Careful '**conversation choreography**' would be needed to ensure an overall **conversation design** that would enable the whole project **community** to engage in **learning** as a **group**. A carefully synchronised **collaboration network** with an efficient **group learning** process would enable the **explicating** of a new **commitment network** which would also be consistent with the new **governing language game**.

7.5.3 Description of the intervention

The situation was different from the previous projects in the following respects:

1. This was a long term intervention requiring the design of a long term conversation. The project commenced in August 1997 and would last at least three months if no problems were encountered.

2. The conversation had to be designed to cope with a large number of sometimes antagonistic individuals with differing needs. In other words, the collaboration network was complex and would have to be carefully synchronized. Conversations would need to be carefully choreographed.

3. Conversation design was explicit - the key participants were exposed to the philosophy of CPD

4. The challenge with previous projects was to design a shared mode of discourse, but this would *not* be a major challenge in this project. An acceptable mode of representation already existed.

The last point needs elaboration. The strategy was to implement and test IMPART dealing and ordering modules. If this could be done immediately, and in such a manner that the project participants could then 'play' with the system in a test environment, then there would be a natural, manipulable, shared space. Instead of talking about dealing, i.e. gathering requirements and communicating functionality etc, participants could interact directly with the system. The project was designed so that all conversation took place via this natural shared space. Three dedicated machines were set up for the duration of the project in a spare office which was called the "Model Office". The idea was that a portfolio manager, a dealer and a data capture clerk would each have an appropriate workstation. Any problems could be seen as a coherent whole by placing orders, trading and assessing the results. People would experience the new dealing system interactively. They would be learning as a group about the system, its settings, and the impact on the whole business process. They would be entering data, looking at each other's screens, learning each other's operations and testing what would happen in certain situations.

The design of the collaboration network was more of a challenge. Appendix E1 shows all the people directly involved in the project. The real network was probably double this size because one person often represented a team. I.T. projects often become unmanageable with such large networks. Appendix E2 shows what happens if each person starts talking to a few others and everyone talks to a kingpin (Roy - the I.T. specialist on this project). How, for example, would this collaboration network remember anything? How would it be synchronised? How could it learn?

In order to design a more manageable collaboration network, the help of the investment manager was needed. Appendix E3 shows how each new project member contributes to a growing "spider web" of

communication channels. Appendix E4 shows options for redesigning this spider web. It shows how individuals can act as links (top right) and how a prototype can act as shared space (bottom left). The investment manager was shown all four diagrams (E1, E2, E3 and E4) and agreed with these design concepts. He agreed that it would be more effective to have one portfolio manager and one dealer in a well synchronised conversation than many dealers and many portfolio managers in a poorly synchronised conversation. I had identified two people to act as 'conversation kingpins', not only for their wide business knowledge, but because of their 'collaborative dispositions'. The investment manager agreed to release these people to the project for a suitable number of hours per week.

Appendix E5 depicts the design of the collaboration network. Each function was represented by 'Solution Builders' who were jointly responsible for implementing the new dealing system. All communication was channelled through these four people. If, for example, Shaleen had a technical problem with the new system, she would *not* talk directly with Roy, but would educate Kobus about the problem. He would then resolve it with Roy, in the shared space provided by the model office. It took a lot of discipline to work in this manner and one had to be conscious of the logic of the conversation design. In the example just given, Kobus would normally pass the query directly to Roy because Kobus is interested in equities, whereas Shaleen is interested in bonds. And Roy would be tempted to pass it on to Fred who is the I.T. dealing expert (as opposed to the IMPART expert). In this manner, Shaleen would naturally end up talking to Fred about her problem. But if this is allowed to happen, a communication spider web begins to grow and it undermines the synchronization mechanism. In particular, learning becomes diffused and the project cannot remember its conversations.

Appendix E5 shows 5 steps:

Step 1.

The solution builders make their best attempt to set up a prototype in the Model Office covering the full ordering, dealing and automated data capture process. They get the prototype to work correctly from a dealing perspective, but know that some settings may be wrong concerning ordering. The conversation for the whole project is thus 'anchored' from a dealing perspective. Thus, if Kobus is unsure of a system setting, he consults his dealing colleagues to get the right answer. Guy, on the other hand, will be more tempted to guess an answer unless he feels that it can be quickly answered by one of his colleagues. Doreen is more concerned with making sure that Kobus and Guy do not 'break' the system. She wants to see that the deals are correctly processed in the IMPART administration system. Roy is the I.T. expert who implements the system settings that the other solution builders decide upon.

Step 2

Ordering experts are invited to the model office to test and correct the ordering process. They are essentially interacting with Guy, but all four solution builders are involved because the ordering process must work across the whole system. For example, Guy may not have catered correctly for separately managed portfolios (SMPs) in step 1. After discussion with Kobus on how the new ordering requirement affects dealing, Roy changes a setting on the ordering module. Ordering now seems to work for SMPs, but when the whole process is run, Doreen sees that it breaks the automated data capture. The conversation involved in tracking and fixing this problem allows the four solution builders to become expert in the whole system *and* the whole business process. These four people can then have viable conversations with the respective colleagues whom they represent because they understand the system and the system's environment.

Step 3

The prototype now has an integrated ordering and dealing process which caters for all ordering and dealing requirements. The solution builders have learned a great deal about both the system and the individual requirements of the colleagues which they represent. The system does not 'break down' at the data capture stage, and as far as Doreen can see it works fine. This step provides a full system test. A whole range of orders and deals is entered on to the system. Data capture experts predict what the outcome should be if they were to capture these deals manually and then compare this with the actual deals that the system updated automatically.

Step 4

It was envisaged that there would be some 'wicked problems'. Software packages tend to constrain the business. They are always programmed for a prototypical operation, whereas all companies tend to develop unique idiosyncrasies. The constraints are usually resisted by the affected individuals. Tough decisions are needed to decide how and where to allocate the pain associated with these constraints. Furthermore, it was known from the outset that the IMPART modules would not be as user-friendly, especially for ordering, as the VB system. The objective of the project was *not* to get a better dealing system, but to improve data accuracy so that the organisation could remain in business. By this stage clients were abandoning the company. It was constantly necessary to remind project members of this objective. Resolving the wicked problems might require changes to the business processes or additional software solutions (but no tailoring of the IMPART modules).

Step 5

Management agreed at the beginning of the project that a genuine 'showstopper' might appear, and that they should be immediately informed about this. If there were no showstoppers, and the wicked problems could be resolved and tolerated, then they would review the complete system, and authorize its release into the 'live' environment.

It was important that the solution builders realise that we were trying to design a synchronised collaboration using a shared mode of discourse in shared space. Although they understood the philosophy behind E5, I grew concerned that they were interpreting it merely as a kind of organisational structure. While E5 did help them to understand their 'roles and responsibilities', I was concerned that they appreciate the synchronization mechanism necessary to make the collaboration viable. E6 shows a metaphoric redescription of the project which was built up interactively with the solution builders. The following question was asked: "How would you describe the *logic* of the project to an outsider?" One answer was: "It's a balancing act, sort of like a seesaw. The Model Office is the seesaw. It hinges on the functionality of the IMPART modules which Roy must install. Kobus stands in the middle with the other dealers on his shoulders and tries to keep the seesaw in balance. But then the Ordering crowd climbs on Guy's shoulders and unbalances the seesaw. Ordering will weigh things down in their favour, but Guy and Kobus will shift their weight to the Data Capture end and help Doreen's crowd on board to restore the balance. Wicked problems are going to crop up, and it will be difficult to get the seesaw completely level at the end of the process, but hopefully both ends will be off the ground." Appendix E6 represents the seesaw metaphor.

E5 represents the design of the collaboration network and E6 provides a metaphoric description of the synchronization mechanism inherent in the collaboration network. The project proceeded according to this design, but at the end of stage 2 a definite showstopper was encountered. The ordering module was grudgingly accepted, but the implications for the dealers were unacceptable. A complex deal authorization procedure introduced too many additional steps into the dealing process. These had the potential of introducing new types of data error. Management then decided to implement the fallback solution, so the VB system was adapted as an interim solution to interface with the IMPART administration system. The conversation design and synchronization mechanism continued to support this new development. The seesaw now hinged on the VB programmer and the VB system. Ordering was unaffected by the amendment to the VB system; from their point of view life would continue unchanged. The Step 3 balancing act could proceed as planned, except it was the VB system which was being tested as opposed to the IMPART dealing modules. The VB system went live with automated data capture of equities in November 1997, and bonds were implemented early in 1998. The same conversation design was used throughout, and the model office was disbanded once bonds were implemented.

The following is a summarised description of the intervention:

1. Forms of life: ordering, dealing, data capture, I.T. software package implementation.
2. Language games: normal public language of asset management, IMPART.
3. Create a manipulable shared space by creating a model office and installing IMPART modules.
4. Design a shared mode of discourse: the model office plus normal asset management language.
5. Design an appropriate collaboration network - limit the potential spider web.

6. Recruit kingpins with a 'collaborative disposition'.
7. Make explicit the conversation design philosophy.
8. Interactively design a logical synchronization mechanism to govern the collaboration network.
9. Metaphorically redescribe the synchronization mechanism to reinforce shared meaning.
10. Govern the conversation over an extended period according to the sychronization mechanism.
11. Live by the seesaw metaphor.

7.5.4 Analysis of the intervention

Project **activity** was **co-ordinated** through a **synchronisation mechanism** that was explicitly **designed** to regulate the **collaboration**. Through **metaphoric redescription**, a 'seesaw' **metaphor** was developed to exemplify the **conversation design** and strengthen the **shared public image** of the manner in which the project would be co-ordinated. This metaphor was explicitly **represented** in order that it become explicitly included as part of the **shared mode of discourse** and assist in **designing** a **synchronization mechanism**. In other words, the **shared mode of discourse** and the **synchronisation mechanism** which **formed** and **governed** the **collaboration network** were designed through explicit '**conversation choreography**'.

The **expressing** of new **private images** was achieved through the medium of the Model Office and IMPART software. This enabled **learning** and **knowledge** to be unfolded in IMPART vocabulary which in turn enabled the **transforming** movement to influence the development of a new **language game** capable of **governing** a new **commitment network**. In other words, the **shared mode of discourse** which enabled **viable conversation** also promoted a vocabulary that enabled the necessary **dialogue** required by the **metaphor** subsystem. As the **metaphor** subsystem influenced the project **community** to develop **shared meaning**, it also enabled the **metaphor** subsystem to influence the **language** subsystem through a **transforming** movement that reflected **shared public images** and **shared purpose** which were expressed in the new IMPART vocabulary.

The **trusting**, **explicating** and **performing** movements began to reinforce each other. The four solution providers **formed** into a cohesive core **community** which was strengthened by their commitment to the **conversation design** and the **group learning** process that allowed them to gain **shared meaning** concerning the whole ordering, dealing and data capture process. This enabled the **trusting** movement to strengthen the relationships within the **collaboration network**. The **collaboration network** was **formed** from an appropriate **shared mode of discourse** that allowed **shared public images** to be clearly **represented** and tested in **shared space**. This enabled the **explicating** movement to establish clear **expectations** of **behaviour** and commitments. Because the Model Office utilised IMPART modules in the **shared space**, the **explicating** movement and the **governing** movement influencing the **commitment network** were mutually reinforcing since they

both employed the IMPART **language game**. The project **community** strengthened its cohesion by **performing** on the commitments it had explicated for itself.

7.6 PROJECT 6 - THE PROPHET PROJECT

7.6.1 Background

This intervention took place in June/July of 1998 in the actuarial department of a major South African Life Assurance Company. Within this department was a team, the Linked Team, who were in the process of implementing a software package called Prophet. This package would allow the valuation of all life policies to be done on a new basis, the embedded value basis, in March 1999. Prophet had to be configured to cope with all life policy business sold or to be sold in the past, present or future. This involved the following skills:

1. A knowledge of the life assurance products.
2. Actuarial understanding of life policy valuation on an embedded value basis.
3. Knowledge of Prophet and how to configure its variables so that it would correctly value a specific product on an embedded value basis.
4. An ability to design independent tests using spreadsheets or Visual Basic so that products implemented on Prophet could be cross-checked for accuracy.
5. An ability to obtain accurate test data.

Initially, Prophet had been poorly received by most of the actuaries. The initial champion for Prophet had long since left the company. A member of the current Linked Team (nicknamed MAP) had developed single-handedly a C++ system to do the entire embedded valuation. MAP's system, called Maverick, worked well but set up a totally unacceptable dependence on MAP who could not be relied upon to stay with the company. The I.T. team, who maintained a mainframe valuation system written in COBOL, were effectively sidelined by Prophet. Their role was reduced to providing life policy data to run against Prophet. The decision to use Prophet, an internationally recognised product, was accepted as being necessary for long term viability, but the history of indifference toward Prophet negated general enthusiasm for the project. Finally, at the time of the intervention, the life assurance company was involved in a takeover by another life assurance company, which I will call "Takeover Company", who were also implementing Prophet. Even though it had been agreed that the March 1999 valuations would remain separate, there was some uncertainty about the longer term future of the whole actuarial function.

The problem was presented by the departmental manger (Adrian), and by the manager of the Linked Team (John). Here is a summary of their perceptions:

1. The team was working very slowly. They did not stick to deadlines, and there was no guarantee that they would meet the March 1999 deadline.

2. The team was 'dysfunctional'. It did not work together as a unit. They did not trust each other and wanted to work autonomously. There were different interpersonal styles within the team which clashed. Project management was already a problem, but would be exacerbated because Jeremy, who was in effect the project manager, would be leaving the company in a few months.

3. The work was perceived to be uninspiring and this caused the low motivation. The big problem was how to make the work more inspiring, but neither manager could see a way to do this.

4. Although there was a slight concern about product knowledge, both managers felt that the Linked Team was capable of doing the project. The team had a viable way of working, and the products that they had implemented were tested for accuracy to the last cent. The team had high quality standards.

5. MAP was a kingpin. The Maverick system was used as the final independent test and MAP was the only one who could run Maverick. MAP was a hard worker, highly intelligent and very effective, but he seldom got to work before 10.00 am, and even though he worked longer hours than most, these late arrivals disrupted the team.

6. Although John was manager of the Linked Team, he was not managing the Prophet project and knew little about Prophet. Because the linked team were required to do other functions no one was dedicated to the Prophet project on a full-time basis. Sometimes a crisis would occur and all work on Prophet would be dropped for as long as two weeks. Project management was weakened by this and it was easy to lose focus on the project.

7. There was an interpersonal conflict between John and Mark. John thought that Mark had always been 'difficult' and that he had often tried to take over the management of the project.

8. The project had no energy and had been drifting for over a year.

The two managers were worried that an important deadline would not be achieved but had run out of ideas as to how to intervene and were basically asking for help.

7.6.2 Framing the intervention

From first perceptions, it seemed that the Prophet project was being fragmented under the combined influence of the **performing**, **trusting** and **explicating** movements. The explicated commitments seemed to emphasise the high quality implementation and testing of products on Prophet and also

adherence to imposed deadlines. It seemed that the **commitment network**, through the **socializing** movement, was able to influence **activity** to meet **expectations** of quality but was unable to socialize team **behaviour** to meet any deadlines. The latter problem, through the **performing** movement, fragmented the project **community**. This in turn generated distrust which negatively influenced the **collaboration network** through the **trusting** movement. Project management seemed to have a significant importance in the Prophet project's **form of life** but the **forming** movement seemed to have great difficulty in establishing any simple, cohesive and binding commitments in this regard. Also, the **normalizing** movement seemed to influence a **shared public image** which regarded project management as 'setting and controlling deadlines'. Similarly, the project management **language game** which was **governing** the **commitment network** seemed to utilise a **final vocabulary** that emphasised 'deadline control'. It seemed that the **transforming** movement would be critical to this intervention because the current project management **language game** seemed incapable of **governing** the **commitment network**. In particular, 'adherence to deadlines' did not seem to be a clearly **shared purpose** .

For these reasons, it would be appropriate to design a **dialogue**. It was necessary to establish the appropriateness of the perceptions of both myself and the two managers. My framing of the intervention might be inadequate and would need to be verified by appreciating the **private images** of the other members of the team. Even if my perceptions were accurate, **dialogue** would still necessary in order to influence the **designing** of a **shared purpose** capable of both **transforming** the **language game** and **governing** a more cohesive project **community**. This would involve a **worldmaking** process directed at unfolding **shared meaning** and creating **shared public images** through **group learning**.

My general impression was that the **regulators** of all three subsystems were weak, including the **conversation** subsystem (i.e. **synchronization mechanism**), but until **learning** had taken place and swept in **multiple representations** of the problem situation, it was not possible to anticipate whether a conversation type of **collaboration** would be needed as part of the intervention.

7.6.3 Description of the Intervention

This intervention was done with CPD in its full and final form. The intervention took place in three steps.

Step 1.
A conversation was designed to promote shared meaning about the issue. A series of one-on-one interviews was conducted with each member of the team. The process was similar to that of Project 2 and Project 4. The purpose was to synthesize shared public image about the situation from individual

private images. The idea was to sweep in as many perspectives on the problem as possible and allow group learning to take place. The comments from the interviews were clustered under topics and can be seen in Appendix F1. Participants were also invited, but not required, to summarize the interviews themselves. They were asked to represent as best they could the conclusions that they had reached during the interview about the Prophet project. Most of the team responded to this request and Appendices F2a to F2f shows thes representations which highlight various issues, ideas and solutions.

F2a suggests that the team should be trusted to organise itself. It shows that the current project process is workable. It argues that if expectations are well managed, management should not worry about the project because they will be informed in good time if expectations seem unattainable.

F2b emphasises the need to be specific about roles, procedures and progress reviews. It details roles to be performed and asks for specific people to be allocated to these roles. It suggests an implementation checklist to be standardized. It asks for regular well organised review sessions.

F2c emphasizes the need for clear understanding of the purpose of the project. It asks for 'carrot and stick' motivators to be clarified and suggests a few design modifications to the project.

F2d asks for clarity on purposes and suggests that a smaller team may have as many pros as a larger team.

F2e emphasizes the difficulties in forecasting deadlines, the testing bottleneck and lack of commitment to deadlines.

F2f highlights the testing bottleneck.

Step 2.
A meeting was arranged with the seven team members and the two managers to review the images that arose from the interviews. Several issues arose in the previous steps as common issues. They were discussed as follows:

1. The two managers were stunned and exasperated by the comments on 'purpose'. They could understand that a few of the newer team members might not appreciate the full context of the project, but it was inconceivable to them that some of the longer-serving were unclear of the exact purpose and consequences of failure. The managers were keen to see how it was possible for there to be any misunderstanding. After a while an exchange took place along the following lines:

TEAM MEMBER 1: The third comment asks 'what is the consequences of not meeting the deadline?' Well, what are they? Never mind the fact that you think we should know this, the point is that as a team we do not know. We are not party to all the discussions that you are. For example, if we do not meet the deadline, why can't we just use the Maverick system as we did last valuation? Why should I take a deadline seriously? Spell it out to us.

DEPARTMENTAL MANAGER: Ok, let me answer the question. When we started this project we were trying to move away from Maverick to Prophet for strategic reasons. We wanted to reduce an unacceptable dependency on MAP by using an internationally recognized package. The deadlines were set in order to speed up the implementation of Prophet which has always been sluggish. I can now see that you re-interpret deadlines according to your own criteria and own calculations. We must avoid this happening again. But now that I am a talking about it, I can see that the situation is different. Because of the takeover there is a great deal of insecurity about the future of our department. The thinking at the moment is that our department will continue to exist and to service our existing policies, at least in the short term. But this is not cast in concrete. This is not a friendly takeover and Takeover Company is also moving to Prophet. We are ahead of them and we need to use our Prophet skills to secure our future. If we cannot deliver or if we fail to make the deadline, I think that would be the end of us as a department. So the consequences of missing the March 1999 valuation deadline would probably result in the whole department being shut down. If we can't do valuations on their terms, why should they keep us?

TEAM MEMBER 2: I can commit to a struggle of keeping our department alive, and therefor of having a secure job. But I found it difficult to relate to a 'strategic deadline' which is basically something that someone has sucked out of their thumb. Maybe we can commit as a team to the purpose of securing the future of the whole department.

TEAM MEMBER 3: I like the idea of being better than Takeover Company on Prophet. We can already beat them on quality; I think we can beat them on speed as well.

This discussion in effect refocused the project. It was agreed that the purpose and language of 'deadlines' was not sufficient to bind the team into a cohesive community, but that the purpose and language of 'survival' was more motivating, meaningful and real.

2. The team as a whole felt that project management issue should be cleared up immediately. It was agreed that the team would decide this as a group, and at another session. This will be described in Step 3.

3. Before the interviews, most people felt that there was too much demand on MAP's time, and they wished he would come to work earlier. Apart from this, they felt that the project was well designed.

During the interviews however, most people focusing on the project design concluded that there was an unacceptable bottleneck in the testing process. This was simply resolved during the group discussion. MAP explained why it was difficult for him to get to work early (a combination of working late and a long commute), but he offered to train another tester. He felt that it would fairly easy to do this and that it would ease the bottleneck considerably. The fact that testing later products was getting easier with increasing implementation would also help to resolve the testing botleneck.

4. The meeting started to run out of time, but during the summary it was felt that progress had been made on most of the issues raised during the interviews. In agreeing to let the team sort out its project management issues, management was exhibiting some support and trust which was lacking before. This plus the clarification of purpose would help motivation. Contrary to initial fears, the team generally did *not* find the project boring, but were demotivated by the negative language surrounding the project. John in particular should be careful that his worrying nature reflected in his talk about the project did not dampen the spirit of the team. Prophet in general had to shrug off its negative connotations and history and be seen as the project which could save the department. There was no time to discuss the recognition issue except to suggest that John become the second tester. In this way he would gain more insight into the project and be in a better position to assess reward and recognition.

Step 3.

During step 2 each team member and the team manager were asked to set aside some time the next day to think about the project management, team roles and team responsibilities. They should take into account the issues raised and synthesized in Step 1 plus the group discussion and agreements of Step 2. They should write down and mail to a central point their detailed ideas on how the project should be managed and who should fulfil specific roles. Once all copies were mailed, the ideas were redestributed so that all the ideas could be viewed by all team members. In this way they could prepare themselves for a group session where they would finalize their decisions. The departmental manager agreed to stay out of the process, and the team manager agreed to be an equal participant in the process.

Two days after the first group discussion of Step 2 the group met again. Each had a copy of their own proposals and everyone else's. Each version of the proposal was numbered and can be seen in Appendix F3a through F3h. After half an hour of discussion, it was decided that option 1, F3a, was first choice and that option 4, F3d, had some important supplements to option 1. Option 1 was used as shared space and MAP's version was utilized as the official record of decisions. As F3a shows, people were free to scribble on their own version of option 1. MAP recorded all final decisions, typed them up neatly, and distributed them that afternoon. Appendix F4 shows the final solution. It shows that F3a has been manipulated and transformed by including aspects of some of the other options, mostly F3d. It will be noted that MAP's role changed from 'Tester' to 'Quality Control' and that he

began to 'oversee' testing rather than doing it all himself. This was made possible by a decision made during the meeting to train all implementers in the preliminary stages of testing on the Maverick system.

It was agreed that it was important for John to take charge of dealing with the remaining outstanding issues, and so the intervention ended at this point.

The following is a brief summary of the intervention:

1. Forms of life: package implementation and testing, actuarial valuation, project management.
2. Language games: Prophet and Maverick, embedded valuation, deadline control.
3. Problem: inability of commitment network to perform on expectations, i.e. missed deadlines.
4. Unfold and reflect on the background of the problem via individual interviews.
5. Individuals off-load their private images into representations of the situation.
6. Synthesize a shared public image by clustering individual private images into shared issues.

7. Reflect on the project design, especially the three sub-system regulators in group discussion:
- shared purpose: from 'strategic deadlines' to 'survival of department'
- language game: from 'worries about deadlines' to 'battle for survival through quality *and* speed'
- synchronization mechanisms: redesigning project roles and alleviating the testing bottleneck.

8. Unfold and off-load private images about project management, roles and responsibilities.
9. Allow time for reflection on each other's various representations.
10. In group discussion choose and agree on the best representation to act as shared space.
11. Manipulate chosen representation by adding parts of other representations.
12. Finalize the design of synchronization mechanisms to govern the collaboration network and to co-ordinate activity.

7.6.4 Analysis of the intervention

As **dialogue** progressed it began **transforming** the **language game governing** the project's **commitment network**. 'Adherence to deadlines' was countered by a more pressing **shared purpose**, namely 'fighting for the continued existence of the department through quality *and* speed'. The latter was a *real* **shared purpose**, whereas the former was considered to be *arbitrary* and unable to influence strong commitment nor exert a **socializing** influence on **activities**. The **dialogue** also unfolded concerns and **multiple representations** regarding a testing bottleneck, project management, roles, and responsibilities which all involved the **explicating** movement. In other words, a clear and cohesive set of commitments needed to be explicated concerning testing, project management, roles and responsibilities. The testing bottleneck also required a better synchronisation of **activity** through the **co-ordinating** movement. This confirmed my initial suspicion that the

regulator of the **conversation** subsystem (i.e. its **synchronization mechanism**) was weak. A conversation type of **collaboration** was needed to derive **cohesive simplicity** from the **multiple representations**, to influence **activities** through the **co-ordinating** movement and to influence the **commitment network** through the **explicating** movement.

Private images were **off-loaded** in the form of written proposals as to how the project should be managed. These detailed the roles, responsibilities and project activity that needed attention. No specially designed **mode of representation** was required as the normal team language was appropriate to enable a **viable conversation**. The team members, who were all actuaries and reasonably proficient in Prophet, were all socialized into the various team **language games** (with the exception of the Maverick system). Neither **expressing** private images nor **off-loading** private images was a problem. However, some form of manipulable **shared space** was needed in order set up the **shared mode of discourse** which is essential to a conversation type of **collaboration**. In a group session, one of the **off-loaded** proposals was chosen through group consensus as the best option for **representing** the **shared public image** of how to deal with the issues. This option became the **shared space** on which the group focused. It was a **manipulable representation** in the sense that images and ideas from the other options were inserted into the chosen option. The chosen option was manipulated in the group session in such a way that each person knew what sentences had been deleted, added, modified, or repositioned. In this manner **a shared mode of discourse** was created that influenced the **forming** of the **collaboration network**. At the end of the process, a final representation contained a **shared public image representing** a synthesis of the original **private images**. The group resolved conflicts between **private images** by manipulating the representation to get a best viable fit between the best private images. The final representation **explicated** clear project **commitments** and included a way of **co-ordinating** project **activity** to eliminate the testing bottleneck.

7.7 CONCLUSION - ANSWERING THE RESEARCH QUESTION

It is now possible to draw conclusions from the 6 action research projects above and to relate the projects so that the learning from the action research can be summarised. The following breakdown gives the fundamental problem and corresponding intervention for each project:

1. The ACT Movement project's fundamental problem was that it employed incompatible **language games** (VANTAGE, I.T. systems analysis and Actuarial) resulting in a breakdown of the shared **interpretative system**. The initial conversation design was dominated by conversation type of **communication**. The intervention consisted of designing two mutually supportive **language games** to repair the **interpretative system** and to use these as **shared space** to support a conversation type of **collaboration**.

2. The Core/Satellite Crisis' fundamental problem was that a peripheral and inappropriate **language game** (Consultant Project Management) sidelined an appropriate and shared **language game** (Asset Management) which resulted in a breakdown of the **interpretative system**. It also was inappropriately dominated by a conversation type of **communication**. The intervention consisted of repairing the **interpretative system** by using the Asset Management **language game** to create a **shared space** to support a conversation type of **collaboration**.

3. The Relational Database project's fundamental problem was that a private **language game** (ERD) was an inappropriate language to use to validate the meaning of a new **language game** (Core/Satellite). The intervention consisted of establishing an **interpretative system** by designing a bridging **language game** (ERD's as spreadsheets) that was then used as **shared space** to support a conversation type of **collaboration**.

4. The Admin Transformation project's fundamental problem was to design an appropriate **collaboration** that would support the synthesis of a mass of **private images** into a cohesive **shared purpose**. The intervention consisted of choosing an appropriate **language game** (PDD) which was suitable as **shared space** to support the **collaboration** and maintain an effective **interpretative system**.

5. The New Dealing System's fundamental problem was designing **synchronization mechanisms** to manage a large **collaboration network**. The intervention consisted of identifying an appropriate **language game** (IMPART modules) and consciously leveraging it to establish a **shared space** (Model Office) that would support the **collaboration**. A disciplined **mode of discourse** was adhered to in order to limit the variety within the **collaboration network**, and this enabled the on-going maintenance of a shared **interpretative system**.

6. The Prophet Project's fundamental problem was to design an appropriate **collaboration** that would support the synthesis of a mass of **private images** into a cohesive **shared purpose**. The intervention consisted of choosing an appropriate **language game** (Written Proposals) and transforming them into a manipulable **shared space** in order to enable **collaboration** and maintain an effective **interpretative system.**

Although each of these projects was very different in nature, by listing the fundamental problem and corresponding intervention in this way it is possible to relate the projects and generalize from one to the other. The common elements central to all the projects are (1) assessment of **language games** (2) designing **shared space** (3) repairing or maintaining a shared **interpretation system** (4) using the previous three elements to support a conversation type of **collaboration**. In addition to this, examination of the detail of the 6 projects shows that the **dialogue** is the route from (1) to (4) and should be included as the 5th fundamental element. This commonality is present in spite of the

various different starting points and nature of the 6 projects. For example, projects 2, 4 and 6 are essentially concerned with a group which is collaborating to make a decision on policy or action. Project 1, 3 and 5 are essentially concerned with ensuring software project success. The commonality of the five elements among the 6 projects supports the argument described on pages 128 to 130 that the CPD process is a journey from 'loose complexity' via 'shared meaning' to 'cohesive simplicity'.

This thesis has developed a new language game for the design of collaborative projects. The vocabulary for this language game is represented as CPD in Fig 3, and has been drawn from the following sources:

1. The philosophy of meaning
2. Systems thinking
3. The sociology of knowledge and social construction of reality
4. Appreciative systems
5. Action research

CPD is thus a synthesis of a number of language games that fold into one another.

Section 2.10 posed the essential research question and it is necessary to assess the degree to which the research has answered this question. Interventions recorded in this chapter were all projects that were either directly related to I.T. or concerned with the organisational environment of I.T. These projects did succeed in promoting collaboration and the success stemmed from an explicit model of the organisational environment in which the interventions took place. CPD achieves collaboration not through analytic techniques but through synthesizing shared meaning and designing shared interpretative systems. It does not attempt to do away with analysis techniques, but attempts to incorporate them into a choreographed and synchronized conversation in such a way that they do not undermine collaboration. Section 6.4 describes the process of using CPD to move from 'loose complexity' to 'cohesive simplicity', and this chapter details interventions where CPD contributed to improving the software process in this way. CPD achieved this by combining systems thinking and language, metaphor and conversation into a systemic lens through which to view an organisation and its projects. All projects entailed metaphoric redescription of the software process; projects were seen as 'organisational conversations' requiring collaborative conversation design rather than 'production lines' to be managed by engineering and analysis.

Section 2.10 also asked whether it would be possible to provide theoretical underpinnings for prior practice. Section 3.5 detailed a prior project, the ACT movement project. It has been shown that the successful intervention based on informal experimentation can be re-interpreted through the theoretical underpinnings which comprise CPD.

CPD represents and exemplifies certain aspects of an organisation. Obviously, there are many other modes of representing organisations and, within the mode of CPD, there could be other valid representations. One could envisage a different arrangement of movements and influencers. This arrangement has come about as an attempt to answer the question posed by Fig 6 (in section 2.10): how can the journey from loose complexity to cohesive simplicity be facilitated by collaborative project design? CPD with its inherent causal influences, appreciative systems framework and its categories for the design of collaborative projects exemplifies an answer to this question.

CHAPTER 8. CONCLUSION AND CRITIQUE

This chapter will summarize the main conclusions of this thesis. It will also attempt to pull together the various strands of argumentation and research into a summarized whole. Finally it will summarize findings and contributions to knowledge.

From a theoretical perspective, this thesis is most closely aligned to the soft systems thinking tradition within the field of systems thinking. The area of application or practical setting for the research has been the software crisis and the ever increasing failure of the software process. The research effort has not been directed toward applying systems thinking directly to the software process itself, but more toward the organizational environment within which the software process operates. Thus the thesis has not become involved in a critical review of current I.T. systems methodologies. Evidence has been provided to show that the software process is more likely to fail as result of lack of effective collaboration rather than because of technological factors. More specifically, the choice of a particular software development methodology is largely irrelevant if the organizational environment within which it is being used is non-collaborative. Thus the thrust of the thesis has not been to develop the ultimate software methodology based on soft systems thinking, but to use a systems approach to promote effective project collaboration and thus a suitable environment within which various software development approaches may be used successfully.

Having said this however, the thesis has on several occasions pointed out situations where employing tools and techniques from standard software methodologies has inhibited collaboration. Indeed, some of the interventions in Chapter 7 consisted of repairing interpretative systems by reducing such tools and techniques (e.g. ERD in Project 3) to the status of private languages because of their inappropriateness at providing the shared language or shared space necessary for collaboration. At the same time, other interventions in Chapter 7 consisted of elevating such private languages (e.g. PDD in Project 4) to the status of a shared language suitable as shared space upon which to found a successful collaboration. The dilemma as to when and how to use the standard tools and techniques of various software development methodologies and their effects on the ability to collaborate has been addressed by introducing theoretical perspectives from the fields of language, metaphor and conversation. These perspectives have been broadly organized by and argued from a position based on the philosophy of meaning rather than on epistemology. This position was heavily influenced by the philosophy of Wittgenstein and the basic position is well captured by the phrase "the change from *knowing* the meaning to *doing* the meaning emphasizes activity" (see page 45). This emphasis is in line with my practitioner approach. Wittgenstein's language games provided a practical way of framing the problem situation. Thus, the current software crisis can be productively viewed as an unresolved struggle between incompatible language games. Specifically, the private language games of I.T., especially software development methodologies, are routinely but inappropriately

legitimized as the public language game for software development projects resulting in a loss of meaning, a breakdown of interpretative systems, disempowerment of business users and consequent inhibition of collaboration. The central finding of this thesis is that this situation can be regularly improved through a conscious attempt to design for collaboration via guidelines provided by the model for the design of collaborative projects (CPD).

The position on meaning and language games is extended by a metaphorical influence which is well captured by Nietzsche's view of language as "a mobile army of metaphors" (see page 67). Rorty's emphasis on contingency in the development of metaphorical language and the ironical view needed with which to reflect on one's final vocabulary combine with the philosophies of Wittgenstein and Nietzsche to push the theoretical positioning of the treatment of language, metaphor and conversation toward a post-modern orientation. The position on the social construction of reality and the systems dynamics of the social process (see Chapter 4) are not grounded in post-modernism yet reflect the contingency which is compatible with a post-modern position. Boulding's preference for the word "image" as opposed to "knowledge" (see page 96) and his key quotation "The study of man is the study of talk. Human society is an edifice spun out of the tenuous webs of conversation" (see page 102) are evidence of this contingency. Based on a central theme of meaning rather than epistemology, the thesis is thus able to synthesise a theoretical position that links the systems approach and the sociology of knowledge to a position on language, metaphor and conversation that reflects a post-modern orientation than is not traditionally associated with soft systems thinking.

The argument that organisations and therefor projects can be metaphorically re-described as conversations is built from this synthesis and gives rise to the idea of conversation design. Four types of conversation are distinguished namely communication, languaging, dialogue and collaboration, and for each an appropriate metaphoric underpinning (or deep metaphor) is developed. An understanding of the four types of conversation and their role in collaboration gives the software practitioner the tool he needs to assess the design of a software project. By understanding the system dynamics of a software project (i.e. Fig 3) and guided by insights into the various conversation types and their role in promoting collaboration (i.e. Table 1), the project designer is empowered to reflect on and re-design the environment within which to make his I.T. methods work effectively. These two sources provide the basis as to whether, for example, a particular tool such as ERD can be used as a shared public language or should be replaced by a new language which may have to be designed from scratch along with corresponding changes to the design of the collaboration network.

Jackson argues that in soft systems thinking "systems are seen as mental constructs of observers rather than entities with a real, objective existence in the world" (Jackson, 1991, page 296). This thesis provides a mental lens for a project designer so, in effect, the system lives in the practitioner's mind. On this basis the thesis is aligned to the soft systems tradition. Jackson also argues of soft

systems thinking that "the immediate aim is to reach 'accommodation' about action to be taken" (ibid., page 296). This focus on accommodation makes soft system thinking a natural source for a practitioner interested in promoting collaboration, yet it is also a source of criticism of soft systems thinking. Jackson elaborates:

"From an emancipatory perspective the soft systems approach seems to be orientated to regulation rather than to radical change. The concern of soft systems thinkers seems to be to understand and facilitate order and cohesion and to seek to preserve the status quo rather than go beyond it. There is a tendency to accept at face value, and work with, existing perceptions of reality. No attempt is made to unmask ideological frames of reference or to uncover the effects of "false-consciousness." Further there is a willingness to take as given "compromises" and "accommodations" within the confines of prevailing power structures. Although developed from an advanced modernist position these criticisms could, of course, be given a post-modernist turn. Post-modernists would criticise soft systems thinkers for their belief that language is a suitable vehicle through which to achieve consensus or accommodation, their belief in progressive "learning", and their failure to take account of the realities of power." (ibid., page 297)

The synthesis of soft systems thinking with post-modern perspectives on language, metaphor and conversation in this thesis go some way to countering the above criticisms. Section 3.3 developed 'metaphoric redescription' as a 6th mode of inquiry to add to Churchman's five designs of inquiring systems. Metaphoric redescription was developed from a post-modern position and, via the transforming movement in Fig 3, has been explicitly linked to language games as an influence to promote organizational transformation. In response to the criticism that soft systems thinking seeks to preserve the status quo, I argue that the process of seeking out deep metaphor, metaphorically re-framing the situation, designing a new project-specific language game and using it to transform and govern a commitment network does address some of the language and power issues raised by Jackson. At a minimum, this can be an *empowering* influence at a micro-level. Project 1 gives an example of this: in the process of metaphorically reframing her task as 'conversation design' rather than 'systems engineering', the systems analyst did transform various language games sufficiently to enable her to cope and to turn software process failure into success. Project 3 gives an example at a macro-level of how lack of attention to the meaning of the two critical key words of a new language game (Core/satellite) had the power to sabotage a mission critical organizational transformation.

This does not imply however that the application of CPD will always and without fail result in effective collaboration. The possibilities for collaboration may be inhibited by structural properties beyond the control of the individuals seeking to collaborate as a group. Structural properties, such as resources available for legitimization, signification and domination, may constrain individual choice and action way beyond the intention and choices of the actors involved in the collaboration. In other words, from a systems perspective, the environment of a software project is itself embedded in a wider organizational environment which may in turn constrain the choices of a group seeking to collaborate. The thesis has highlighted the importance of selecting or designing a suitable mode of discourse and Fig 3 has included the legitimizing movement to sensitise the practitioner to the dynamics involved in

legitimizing a mode of discourse. The thesis has also highlighted that those in power tend to control the language games and modes of discourse. A group seeking to collaborate may be confronted by language games and modes of discourse which inhibit collaboration but which cannot be transformed, re-designed or rendered illegitimate because of power dynamics beyond the control of the collaborating group. When such structural mediation and power dynamics are combined with the contingency inherent in 'the tenuous webs of conversation' it must be acknowledged that CPD can never be used a recipe to guarantee collaboration. CPD is an idealised model to hold in the mind in order to intervene in such a way as to *encourage the emergence of collaboration*. A particular collaboration may never succeed at overcoming the power dynamics that would allow it to transform an inhibitive language game or legitimize a more appropriate mode of discourse. However, at a micro level, a practitioner may still be *empowered* in such a situation by an appreciation of the project dynamics as offered by Fig 3. A close reading of the interventions described in Chapter 7 will reveal that the projects succeeded not by challenging and attempting to discredit the legitimized modes of discourse, but by building temporary bridging languages that enabled the practitioner to cope in a problematic situation. For example, Project 3 built a temporary mode of discourse (ERD as spreadsheets and a mode of talking about them using colour coding) which built a bridge between three powerful and legitimized language games (asset management language, Core/Satellite language and ERD). An appreciation of the system dynamics offered by Fig 3 and insights provided by Table 1 *empowered* the interventionist to cope and encouraged collaboration to emerge.

This thesis has attempted to fill a hole in literature. The literature research found are no texts which combine the following:

1. The software process as a collaborative project
2. A systemic approach to designing collaborations
3. A systemic understanding of language, metaphor and conversation and their roles in collaboration

In fact, it is extremely difficult to find texts which have an adequate treatment of each of these topics individually. This research is thus a synthesis of positions found in all the above categories, but in order to do so, the thesis had to establish positions on each of the above topics from first principles with little help from outside texts. There is much literature on the software process itself but very little on the *environment* of the software process which is a fundamental departure point for this thesis. Another difficulty is that the I.T. literature tends to separate project management from the software process. There are, for example, many texts on project management and many texts on systems analysis and design. However, there are much fewer texts that concern themselves with project managing systems analysis and design, but these texts then proceed to *ignore the sociology* of such projects. When the few texts on the sociology of software projects are encountered (for example Peopleware, DeMarco, 1987) they are often treated in isolation from the software process. Not only are project management, the software process and the sociology of software projects are poorly integrated in current I.T. literature, but the focus is on analytic project *management* as opposed to creative project *design*. In other words, current I.T. literature gives the impression that projects can be

well designed through analytic techniques such as critical path analysis and that management is the main issue. This thesis challenges this assumption and therefor makes a distinction between project management and project design which is not apparent in I.T. literature. The thesis makes no attempt to discredit traditional I.T. project management, but does highlight the point that a well managed project may still be poorly designed and therefore fail. There is no coherent body of literature covering the creative design of software projects. Checkland and Holwell point out that scholarly I.T. journals such as MIS Quarterly hold "unquestioned editorial assumptions of a deeply positivistic kind" (Checkland and Holwell, 1998, page 52) which is incompatible with research that assumes that social reality is continually constructed and re-constructed.

The following summarizes this thesis' main contributions to knowledge:

1. Research method - The action research method utilizes a declared-in-advance framework in the form of an influence diagram which is progressively evolved to become a final product (Fig 3). This method of research will be able to be used as a model for other research projects. This method provides an example of how to combine theoretical research with practical research. The manner in which the theoretical position on the philosophy of meaning has been progressively developed into a practical tool for the interventionist is evidence of this.

2. Theoretical grounding of collaboration - Collaboration has become topical in business but the supporting literature is shallow with regard to its theoretical depth and shows little evidence of comprehensive theoretical grounding. This thesis has made a thorough attempt to provide a theoretical grounding for collaboration. The categories for the design of collaborative products as summarized in Table 1 are evidence of this.

3. Metaphoric redescription - By focusing on the philosophy of meaning, the thesis has built a bridge between soft systems thinking and post-modernism allowing certain criticisms of soft systems thinking to be countered. By developing metaphoric redescription from postmodern perspectives on language, metaphor and conversation, a new mode of inquiry, a new source of empowerment and a new source for organizational transformation have been contributed to the interventionist's repertoire.

4. Re-framing the software process - The thesis has contributed a systems approach to reframing the software process and used systems principles plus metaphoric redescription to provide new insights into the software process. The deep metaphor of 'engineering' has been challenged and a re-framing allows software projects to be viewed as 'conversations' which need careful design. Associated with this is the notion that the software process is a journey from 'loose complexity' via 'shared meaning' to 'cohesive simplicity'. The model building resulting in Fig 3 and Table 1 has been designed to support this journey.

5. A model for the design of collaborative projects (CPD) - This model contributes an original diagnostic lens with which to understand the social process within organizations. It is also an empowering lens for the project designer. This makes a particular contribution to the field of project management and provides it with a new language game to supplement the conventional language game (i.e. critical path analysis, Gantt Charts, etc.). In particular, it highlights the importance of conversation design and provides practical insight on how to achieve viable conversation design within a collaborative project.

6. The environment of I.T. - CPD provides a means of empowering the I.T. practitioner by providing guidelines on how to intervene in the environment of the software process in order to increase the probability of software process success. It provides the missing element that software developers require, namely a deep understanding on how to promote the collaboration which is so critical to the software process. In this manner, the thesis makes its contribution to alleviating the software crisis.

CHAPTER 9. REFERENCES

Ackoff, Russell L.

Creating the Corporate Future

1981 John Wiley & Sons, New York

Ackoff, Russell L.

The Democratic Corporation

1994 Oxford University Press, Oxford

Argyris, C., Putnam, R. and McLain-Smith, D.

Action Science: concepts methods and skills for research and intervention

1982 Jossey-Bass, San Francisco

Bennett, Roger and Oliver, Jim

How to Get the Best from Action Research - A Guidebook

1988 MCB University Press Ltd, Bradford

Berger, Peter and Luckmann, Thomas

The Social Construction of Reality

1966 Penguin Books Ltd, London

Bohm, David

On Dialogue

1996 Routledge, New York

Boulding, Kenneth E.

The Image: Knowledge in life and society

1956 Ann Arbor Paperbacks, The University of Michigan Press

Capra, Fritjof

The Web of Life: A new synthesis of mind and matter

1996 HarperCollinsPublishers, London

Checkland, Peter

Systems Thinking, Systems Practice

1981 John Wiley & Sons, Chichester

Checkland, Peter & Holwell, Sue

Information, Systems and Information Systems: Making Sense of the Field

1998 John Wiley & Sons, Chichester

Churchman, C. West

The Systems Approach and its Enemies

1979 Basic Books Inc. New York

Churchman, C.West

The Design of Inquiring Systems: Basic concepts of systems and organization

1971 Basic Books, Inc. New York

Collins, Tony with Bicknell, David

Crash - Learning from the world's worst computer disasters

1998 Simon & Schuster Ltd, London

DeMarco, T. & Lister, T.

Peopleware: Productive projects and teams

1987 Dorset House Publishing Company, New York, NY

DeMarco, Tom

Controlling Software Projects: Management, measurement & estimation

1982 Yourdon Press, Prentice Hall PTR, Engelwood Cliffs, New Jersey

DeMarco, Tom

Why Does Software Cost So Much?

1995 Dorset House Publishing, New York

Dennett, Daniel

Kinds of Minds: Toward an understanding of consciousness

1996 Basic Books, New York

Field, Tom

When bad things happen to good projects

1997 Computing SA, November 1997

Finch, Henry LeRoy

Wittgenstein

1995 Element Books Inc, Rockport MA

Flood, R. L. & Jackson, M. C.

Creative Problem Solving: Total Systems Intervention

1991 John Wiley & Sons, Chichester

Functional Analysis of Office requirements - A multiperspective approach

Schafer, G

1988 John Wiley & Sons

Goodman, Nelson

Ways of World Making

1978 Hackett Publishing Company, Indianapolis

Grayling, A.C.

Wittgenstein (Past Masters)

1988 Oxford University Press, Oxford

Jackson, Michael C.

Modernism, Post-Modernism and Contemporary Systems. In: Critical Systems Thinking: Directed readings

1991 Edited by Flood, R.L. & Jackson M.C., John Wiley & Sons, Chichester

Kauffman, Draper L

Systems One: An Introduction to Systems Thinking. The innovative learning series

1980 Future Systems Inc, Minneapolis

Kauffman, Walter (Editor)

The Portable Nietzsche

1954 The Viking Press, New York

Lakoff, George & Johnson, Mark

Metaphors We Live By

1980 The University of Chicago Press

Lewin, Kurt

Action Research and Minority Problems in Resolving Social Conflicts: selected papers on group dynamics

1946 American Psychological Association, Washington (1997)

Longworth, G. & Nicholls, D.

SSADM Manual: Volume 2 - Techniques and Documentation

1986 NCC Publications, Manchester

Lyotard, Jean-Francois

The Postmodern Condition: a report on knowledge

1979 University of Minnesota Press, Minneapolis

Martin, James

Recommended Diagramming Standards For Analysts & Programmers: A basis for automation

1987 Prentice-Hall Inc. Englewood Cliffs, New Jersey

Mason, R. O. and Mitroff, I. I

Challenging Strategic Planning Assumptions

1981 Wiley, New York

Maturana, Humberto R. & Varela, Francisco J.

The Tree of Knowledge: The biological roots of understanding

1987 Shambhala Publications Inc, Boston

McLeod, Graham & Smith, Derek

Managing Information Technology Projects

1996 Boyd & Fraser Publishing Company, Danvers

Morgan, Gareth

Images of Organisation

1997 Sage Publications Inc, Thousand Oaks, California

Patching, David

Practical Soft Systems Analysis

1990 Pitman Publishing, London

Reddy, M.J.

The Conduit Metaphor: A case of frame conflict in our language about language

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Rorty, Richard

Contingency, Irony and Solidarity

1989 Cambridge University Press, Cambridge

Schon, Donald A.

Generative metaphor: A perspective on problem-setting in social policy

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Schrage, Michael

No More Teams! Mastering the Dynamics of Creative Collaboration

1995 Bantam Doubleday Dell Publishing Group, New York

Senge, P. M., Roberts, C., Ross R.B., Smith, B.J. & Kleiner, A.

The Fifth Discipline Fieldbook

1994 Currency, Doubleday, New York

Senge, Peter M.

The Fifth Discipline: The art and practice of the learning organization

1990 Century Business, London

Singer Jr, E. A.

In Search of a Way of Life

1945 Columbia University Press, New York

Standish Group, The

Chaos

1995 Internet adress: http://www.standishgroup.com.html

Stern, J.P.

Nietzsche (Modern Masters)

1978 Fontana Press, London

Ulrich, Werner

Critical Heuristics of Social Planning

1983 John Wiley & Sons, Chichester

Vickers, Geoffrey

Value Systems and Social Process

1968 Tavistock Publications Ltd, London

Vickers, Geoffrey

Freedom in a Rocking Boat

1970 Allen Lane, The Penguin Press, London

Weinberg, Gerald M

Rethinking Systems Analysis and Design

1988 Dorset House Publishing, New York

Whitten, Jeffrey L., Bently, Lonnie D. & Barlow, Victor M.

Systems Analysis & Design Methods (Third edition)

1994 Richard D. Irwin, Inc, Burr Ridge, Illinois

Winograd, Terry & Flores, Fernando

Understanding Computers and Cognition

1986 Addison-Wesley Publishing Company, Norwood NJ

Wittgenstein, Ludwig

Philosophical Investigations

1958 Prentice Hall, Englewood Cliffs, New Jersey

Wittgenstein, Ludwig

On Certainty

1969 Blackwell Publishers Ltd, Oxford

Yourdon, Ed

On Emerging Software Technologies

1991 Conference, Johannesburg (Organisers Mike Bergen & Associates)

CHAPTER 10. BIBLIOGRAPHY

Philosophy

Ackoff, Russell L. & Churchman, C. West

Methods of Inquiry: An Introduction in Philosophy and Scientific Method

1950 Education Publishers, Saint Louis

Andersen, Walter Truett

The Fontana Postmodernism Reader

1996 Fontana Press, London

Berger, Peter and Luckmann, Thomas

The Social Construction of Reality

1966 Penguin Books Ltd, London

Bernstein, Richard J.

Beyond Objectivism and Relativism: Science, Hermaneutics, and Praxis

1983 University of Pennsylvania Press, Philadelphia

Bernstein, Richard J.

The New Constellation: The ethical-political horizons of Modernity/Postmodernity

1991 The MIT press, Cambridge Massachusetts

Blackburn, Simon

The Oxford Dictionary of Philosophy

1994 Oxford University Press, Oxford

Finch, Henry LeRoy

Wittgenstein

1995 Element Books Inc, Rockport MA

Goodman, Nelson

Ways of World Making

1978 Hackett Publishing Company, Indianapolis

Grayling, A.C.

Wittgenstein (Past Masters)

1988 Oxford University Press, Oxford

Honderich, Ted

The Oxford Companion to Philosophy

1995 Oxford University Press, Oxford

Kauffman, Walter (Editor)

The Portable Nietzsche

1954 The Viking Press, New York

Lyotard, Jean-Francois

The Postmodern Condition: a report on knowledge

1979 University of Minnesota Press, Minneapolis

Magee, Bryan

The Great Philosophers: An introduction to Western philosophy

1987 Oxford University Press, Oxford

Nietzsche, Friedrich (Translated by: R. J. Hollingdale)

Thus Spoke Zarathustra

1961 Penguin Books, London

Pears, David

Wittgenstein (Modern Masters)

1971 Fontana Press, London

Rorty, Richard

Contingency, Irony and Solidarity

1989 Cambridge University Press, Cambridge

Russell, Bertrand

The Problems of Philosophy

1988 Prometheus Books, Buffalo, New York

Scruton, Roger

Kant (Past Masters)

1982 Oxford University Press, Oxford

Scruton, Roger

Modern Philosophy: An introduction and survey

1994 Mandarin Paperbacks, London

Sim, Stuart

Jean Francois Lyotard

1996 Prentice-Hall/Harvester Wheatsheaf, Hemel-Hempstead

Singer Jr, E. A.

In Search of a Way of Life

1945 Columbia University Press, New York

Singer Jr, E. A. (Edited by C. West Churchman)

Experience and Reflection

1959 University of Pennsylvania Press, Philadelphia

Stern, J.P.

Nietzsche (Modern Masters)

1978 Fontana Press, London

Tanner, Michael

Nietzsche (Past Masters)

1994 Oxford University Press, Oxford

Tarnas, Richard

The Passion of The Western Mind: Understanding the ideas that have shaped our world view

1991 Pimlico, London

Wittgenstein, Ludwig

Philosophical Investigations

1958 Prentice Hall, Englewood Cliffs, New Jersey

Wittgenstein, Ludwig

On Certainty

1969 Blackwell Publishers Ltd, Oxford

Systems Theory & Philosophy

Ackoff, Russell L.

Creating the Corporate Future

1981 John Wiley & Sons, New York

Ackoff, Russell L.

The Democratic Corporation

1994 Oxford University Press, Oxford

Bateson, Gregory

Mind and Nature: A Necessary Unity

1979 Bantam Books, New York

Bohm, David

Thought as a system

1994 Routledge, London

Bohm, David

On Dialogue

1996 Routledge, New York

Boulding, Kenneth E.

The Image: Knowledge in life and society

1956 Ann Arbor Paperbacks, The University of Michigan Press

Bowler, T. Downing

General Systems Thinking: Its scope and applicability

1981 Elsevier North Holland, Inc. New York

Capra, Fritjof

The Turning Point: Science, society and the rising culture

1982 Flamingo, London

Capra, Fritjof

The Web of Life: A new synthesis of mind and matter

1996 HarperCollinsPublishers, London

Checkland, Peter

From Optimizing to Learning: A development of systems thinking for the 1990s in Critical Systems Thinking: Directed readings

1991 Edited by Flood, R.L. & Jackson M.C., John Wiley & Sons, Chichester

Churchman, C. West

The Systems Approach and its Enemies

1979 Basic Books Inc. New York

Churchman, C.West

The Design of Inquiring Systems: Basic concepts of systems and organization

1971 Basic Books, Inc. New York

Churchman, C.West

Thought and Wisdom

1982 Intersystems Publications, Seaside California

Jackson, Michael C.

Social Sytsems Theory and Practice: The need for a Critical Approach in Critical Systems Thinking: Directed readings

1991 Edited by Flood, R.L. & Jackson M.C., John Wiley & Sons, Chichester

Jackson, Michael C.

Modernism, Post-Modernism and Contemporary Systems. In: Critical Systems Thinking: Directed readings

1991 Edited by Flood, R.L. & Jackson M.C., John Wiley & Sons, Chichester

Kauffman, Draper L

Systems One: An Introduction to Systems Thinking. The innovative learning series

1980 Future Systems Inc, Minneapolis

Keeney, Bradford

The Aesthetics of Change

1983 The Guilford Press, New York

Maturana, Humberto R. & Varela, Francisco J.

The Tree of Knowledge: The biological roots of understanding

1987 Shambhala Publications Inc, Boston

Midgley, Gerald

What is this thing called CST? (in Critical Systems Thinking: Current Research and Practice)

1996 Flood, R.L. & Romm, N.R.A. (editors), Plenum Press, New York

Midgley, Gerald

The Ideal of Unity and the Practice of Pluralism in Systems Science (in Critical Systems Thinking: Current Research and Practice)

1996 Flood, R.L. & Romm, N.R.A. (editors), Plenum Press, New York

Mitroff, Ian & Linstone, Harold

The Unbounded Mind: Breaking the chains of traditional business thinking

1993 Oxford University Press, New York

Tenner, Edward

Why Things Bite Back: Predicting the problems of progress

1996 Fourth Estate Ltd, London

Ulrich, Werner

Critical Heuristics of Social Planning

1983 John Wiley & Sons, Chichester

Vickers, Geoffrey

Value Systems and Social Process

1968 Tavistock Publications Ltd, London

Vickers, Geoffrey

Freedom in a Rocking Boat

1970 Allen Lane, The Penguin Press, London

Systems Practice

Allee, Verna

The Knowledge Evolution: Expanding Organisational Intelligence

1997 Butterworth-Heinemann, Boston

Bignell, Victor & Fortune, Joyce

Understanding Systems Failures

1984 Manchester University Press, Manchester

Checkland, Peter

Systems Thinking, Systems Practice

1981 John Wiley & Sons, Chichester

Flood, R. L. & Jackson, M. C.

Critical Systems Thinking: Directed Readings

1991 John Wiley & Sons, Chichester

Flood, R. L. & Jackson, M. C.

Creative Problem Solving: Total Systems Intervention

1991 John Wiley & Sons, Chichester

Flood, Robert L.

Solving Problem Solving: A Potent Force for Effective Management

1995 John Wiley & Sons, Chichester

Flood, Robert L. and Romm, Norma R. A. (Editors)

Critical Systems Thinking: Current research and practice

1996 Plenum Press, New York

Jackson, M.C

Towards a Coherent Pluralism in Management Science: Working Paper No 16

1997 Lincoln School of Management Working Paper Series

Mason, R. O. and Mitroff, I. I

Challenging Strategic Planning Assumptions

1981 Wiley, New York

Patching, David

Practical Soft Systems Analysis

1990 Pitman Publishing, London

Richmond, Barry

Systems Dynamics/Systems Thinking: Let's Just Get On With It

1994 International Sytems Dynamics Conference, Sterling, Scotland

Senge, P. M., Roberts, C., Ross R.B., Smith, B.J. & Kleiner, A.

The Fifth Discipline Fieldbook

1994 Currency, Doubleday, New York

Senge, Peter M.

The Fifth Discipline: The art and practice of the learning organization

1990 Century Business, London

Strümpfer, Johan

Thinking and Action: The Systems Challenge and Response

1996 Systems Conference, University of Cape Town

Strümpfer, Johan

Ensuring Delivery Through Organisational Change

1997 International Systems Thinking Conference, University of Cape Town

Sweeney, Linda Booth & Meadows, Dennis

The Systems Thinking Playbook (Volume 2)

1966 Linda Booth Sweeney

Ulrich, Werner

Critical Heuristics of Social Systems Design in Critical Systems Thinking: Directed readings

1991 Edited by Flood, R.L. & Jackson M.C., John Wiley & Sons, Chichester

Watzlawick, P., Weakland, J. & Fisch, R.

Change: Principles of problem formation and problem resolution

1988 W.W. Norton and company, New York

Watzlawick,P., Bavelas, J. B. & Jackson, D. D.

Pragmatics of Human Communication: A study of interactional patterns, pathologies and paradoxes

1967 W. W. Notron and company, New York

Systems Practice & Software Process

Abdel-Hamid, Tarek & Madnick, Stuart E.

Software Project Dynamics: An integrated approach

1991 Prentice Hall, Englewood Cliffs, N.J.

Checkland, Peter & Holwell, Sue

Information, Systems and Information Systems: Making Sense of the Field

1998 John Wiley & Sons, Chichester

Functional Analysis of Office requirements - A multiperspective approach

Schafer, G

1988 John Wiley & Sons

Weinberg, Gerald M

Rethinking Systems Analysis and Design

1988 Dorset House Publishing, New York

Weinberg, Gerald M.

Quality Software Management: Volume 1 - Systems Thinking

1992 Dorset House Publishing, New York

Winograd, Terry & Flores, Fernando

Understanding Computers and Cognition

1986 Addison-Wesley Publishing Company, Norwood NJ

Wood-Harper, A. T. & Antill, L. & Avison, D. E.

Information Systems Definition: The Multiview Approach

1985 Blackwell Scientific Publications

Software Process

Collins, Tony with Bicknell, David

Crash - Learning from the world's worst computer disasters

1998 Simon & Schuster Ltd, London

Cusumano, Michael A. & Selby, Richard W.

Microsoft Secrets: How the world's most powerful software company creates technology, shapes markets, and manages people

1995 HarperCollinsPublishers, London

DeMarco, T. & Lister, T.

Peopleware: Productive projects and teams

1987 Dorset House Publishing Company, New York, NY

DeMarco, Tom

Controlling Software Projects: Management, measurement & estimation

1982 Yourdon Press, Prentice Hall PTR, Engelwood Cliffs, New Jersey

DeMarco, Tom

Why Does Software Cost So Much?

1995 Dorset House Publishing, New York

DeMarco, Tom

Deadline: A novel about project management

1997 Dorset House Publishing, New York

Fisher, Alan S.

CASE: using software development tools

1991 John Wiley & Sons Inc, New York

Graham, Ian

Migrating to Object Technology

1995 Addison-Wesley Publishing Company, Wokingham

Humphrey, Watts S

Managing the Software Process

1989 Addison-Wesley, Reading, Mass.

Kosko, Bart

Fuzzy Thinking: The new science of fuzzy logic

1993 Flamingo, London

Longworth, G. & Nicholls, D.

SSADM: Volume 1 - Tasks and Terms

1986 NCC Publications, Manchester

Longworth, G. & Nicholls, D.

SSADM Manual: Volume 2 - Techniques and Documentation

1986 NCC Publications, Manchester

Martin, James

Recommended Diagramming Standards For Analysts & Programmers: A basis for automation

1987 Prentice-Hall Inc. Englewood Cliffs, New Jersey

Martin, James

Information Engineering, Volume 2: Strategy and Analysis

1989 Prentice Hall, Engewood Cliffs, N.J.

Martin, James

Information Engineering, Volume 1: Introduction and Principles

1989 Prentice Hall, Engewood Cliffs, N.J.

Martin, James & McClure, Carma

Action Diagrams: Clearly structured specifications, programs, and procedures

1985 Prentice Hall, Engewood Cliffs, N.J.

McLeod, Graham & Smith, Derek

Managing Information Technology Projects

1996 Boyd & Fraser Publishing Company, Danvers

McNeill, Daniel & Freiberger, Paul

Fuzzy Logic: The revolutionary computer technology that is changing our world

1993 Touchstone, New York

Standish Group, The

Chaos

1995 Internet adress: http://www.standishgroup.com.html

Whitten, Jeffrey L., Bently, Lonnie D. & Barlow, Victor M.

Systems Analysis & Design Methods (Third edition)

1994 Richard D. Irwin, Inc, Burr Ridge, Illinois

Yourdon, Ed

On Emerging Software Technologies

1991 Conference, Johannesburg (Organisers Mike Bergen & Associates)

Creativity

De Bono, Edward

The Happiness Purpose

1977 Penguin Books, London

De Bono, Edward

Serious Creativity: Using the power of lateral thinking to create new ideas

1992 Harper Collins Publishers, London

Hargrove, Robert

Mastering the Art of Creative Collaboration

1998 McGraw-Hill, New York

Michalko, Michael

Thinkertoys: A handbook of business creativity for the 90s

1991 Ten Speed Press, Berkeley, California

Schrage, Michael

No More Teams! Mastering the Dynamics of Creative Collaboration

1995 Bantam Doubleday Dell Publishing Group, New York

von Oech, Roger

A Whack on the Side of the Head: How you can be more creative

1983 Thorsons, London

Language and metaphor

Austin, J.L.

Performative Utterances (1961)

1996 in The Philosophy of Language, Martinich, A.P. (Editor), Oxford University Press

Black, Max

More About Metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Green, T.

Learning Without Metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Grice, H.P.

Logic and Conversation (1975)

1996 inThe Philosophy of Language, Martinich, A.P. (Editor), Oxford University Press

Lakoff, George

The Contemporary Theory of Metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Lakoff, George & Johnson, Mark

Metaphors We Live By

1980 The University of Chicago Press

Levin, Samual R.

Language, Concepts and Worlds : Three domains of metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Martinich, A.P. (Editor)

The Philosophy of Language

1996 Oxford University Press, New York

Morgan, Gareth

Images of Organisation

1997 Sage Publications Inc, Thousand Oaks, California

Ortony, Andrew

Metaphor, Language and Thought

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Ortony, Andrew (Editor)

Metaphor and Thought

1993 Cambridge University Press, Cambridge

Petrie, H.G. & Oshlag, R.S.

Metaphor and Learning

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Reddy, M.J.

The Conduit Metaphor: A case of frame conflict in our language about language

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Schon, Donald A.

Generative metaphor: A perspective on problem-setting in social policy

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Searle, John R.

What is a Speech Act? (1965)

1996 inThe Philosophy of Language, Martinich, A.P. (Editor), Oxford University Press

Searle, John R.

A Taxonomy of Illocutionary Acts (1979)

1996 in: The Philosophy of Language, Martinich, A.P. (Editor), Oxford University Press

Searle, John R.

Indirect Speech Acts (1975)

1996 inThe Philosophy of Language, Martinich, A.P. (Editor), Oxford University Press

Serale, John R.

Metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Steinberg, R.J., Tourangeau, R. & Nigro, G.

Metaphor, Induction and social policy: The convergence of macroscopic and microscopic views

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Sticht, Thomas

Educational Uses of Metaphor

1993 in Metaphor and Thought, Andrew Ortony (editor), Cambridge University Press, Cambridge

Vygotsky, Lev

Thought and Language

1986 The MIT Press, Cambridge, Massachusetts

Watzlawick, Paul

The Language of change: Elements of therapeutic Communication

1978 W.W. Norton & Company, New York

Way, Eileen

Knowledge Representantion and Metaphor

1991 Kluwer Academic Publishers, Dordrecht

General

Argyris, C., Putnam, R. and McLain-Smith, D.

Action Science: concepts methods and skills for research and intervention

1982 Jossey-Bass, San Francisco

Beck, Don E. & Cowan, Christopher C.

Spiral Dynamics: Mastering values, leadership, and change

1996 Blackwell Publishers Inc., Cambridge, Massachusetts

Bennett, Roger and Oliver, Jim

How to Get the Best from Action Research - A Guidebook

1988 MCB University Press Ltd, Bradford

Champy, James

Reengineering Management: The mandate for new leadership

1995 HarperCollinsPublishers

Covey, Stephen R.

The Seven Habits of Highly Effective People

1989 Simon & Schuster, London

Covey, Stephen R.

Principle Centered Leadership: a philosophy for life and success in business

1990 Simon & Schuster, London

Dennett, Daniel

Kinds of Minds: Toward an understanding of consciousness

1996 Basic Books, New York

Dryden, G. & Vos, J.

The Learning Revolution: A lifelong learning programme for the wolrd's finest computer: your amazing brain

1994 Accelerated Learning Systems, Aylesbury UK

Gibson, Rowan (Editor)

Rethinking the Corporate Future

1996 Nicholas Brealey Publishing Ltd, London

Goldratt, Eliyahu M.

The Goal: A process of ongoing improvement

1984 National Productivity Institute, Pretoria

Hammer, Michael & Champy, James

Reengineering the Corporation

1993 Nicholas Brealey Publishing, London

Handy Charles

The Empty Raincoat

1994 Arrow Books Ltd

Handy, Charles

The Age of Unreason

1989 Arrow Books Ltd

Handy, Charles

Beyond Certainty

1995 Arrow Books Ltd

Handy, Charles

The Hungry Spirit

1998 Arrow Books Ltd, London

Lewin, Kurt

Action Research and Minority Problems in Resolving Social Conflicts: selected papers on group dynamics

1946 American Psychological Association, Washington (1997)

Lewin, Kurt

Resolving Social Conflicts & Field Theory in Social Science

1997 American Psychological Association, Washington DC

Lynch, Dudley & Kordis, Paul L.

The Strategy of the Dolphin: Winning elegantly by coping in a world of turbulent change

1988 Arrow Books Ltd

McRae, Hamish

The World in 2020: Power, culture and prosperity: a vision of the future

1994 HarperCollinsPublishers, London

Naisbitt, John

Global Paradox

1994 Nicholas Brealey Publishing, London

Naisbitt, John and Aburdene, Patricia

Megatrends 2000

1990 Pan Books Ltd, London

Pascale, Richard

Managing on the Edge: How successful companies use conflict to stay ahead

1990 Penguin Books, London

Peters, Tom

Liberation Management: Necessary disorganisation for the nanosecond nineties

1992 MacMillan, London

Sasdar, Ziauddin and Abrams, Iwona

Chaos For Beginners

1998 Icon Books Ltd, Cambridge

Semler, Ricardo

Maverick

1993 Arrow Books Limited, London

Silbiger, Steven

The 10-Day MBA

1993 Judy Piatkus (publishers) Ltd

Toffler, Alvin

Powershift: Knowledge, wealth and violence at the edge of the 21st century

1990 Bantam Books, New York

Wheatley, Margaret J

Leadership and the New Science: Learning about organisation from an orderly universe

1992 Berret-Koehler Publishers, San Francisco

Wycoff, Joyce (with Richardson, Tim)

Transformation Thinking

1995 Berkley Books, New York

Journals

Apgar IV, Mahlon

The Alternative Workplace: Changing where and how people work

1998 Harvard Business Review, May-June 1998

Argyris, Chris

Teaching Smart People How to Learn

1991 Harvard Business Review, May-June 1991

Argyris, Chris

Good Communication That Blocks Learning

1994 Harvard Business Review, July-August, 1994

Argyris, Chris

Empowerment: The Emperor's New Clothes

1998 Harvard Business Review, May-June 1998

Bartlett, C. A. & Ghoshal, S.

Changing the Role of Top Management: Beyond Systems to People

1995 Harvard Business Review, May-June 1995

Bashein, B.J & Markus, M.L.

A Credibility Equation for IT Specialists

1977 Sloan Management Review, Summer 1977

Boddie, John

Managing from the Middle

1997 Software Development, September 1997

Campbell, A. & Alexander, M.

What's Wrong With Strategy?

1997 Harvard Business Review, November-December, 1997

Conger, Jay

The Necessary Art of Persuasion

1998 Harvard Business Review, May-June 1998

Constantine, Larry

Real Life Requirements

1998 Software Development, May 1998

Cusumano, Michael A.

How Microsoft Makes Large Teams Work Like Small Teams

1977 Sloan Management Review, Fall 1997

Davenport, T.H., De Long, D. & Beers, M.C.

Successful Knowledge Management Projects

1998 Sloan Management Review, Winter 1998

Davenport, T.H., Jarvenpaa, S.L., & Beers M.C.

Improving Knowledge Work Processes

1996 Sloan Management Review, Summer 1996

Davenport, Thomas H.

Putting the Enterprise into the Enterprise System

1998 Harvard Business Review, July - August 1998

de Geus, Arie

The Living Company

1997 Harvard Business Review, March-April 1997

Drucker, P.F., Dyson, E., Handy, C., Saffo, P. & Senge, P.M.

Looking Ahead: Implications of the present

1997 Harvard Business Review, September - October, 1997

Eisenhardt, K.M., Kahwajy, J.L., Bourgeois III, L.J.

How Management Teams Can Have a Good Fight

1997 Harvard Business Review, July-August 1997

Elton, J. & Roe, J.

Bringing Discipline to Project Management

1998 Harvard Business Review, March-April, 1998

Emery, James

Adaptive Software Development

1998 Software Development, April 1998

Euske, K.J., & Player, S.R.

Leveraging Management Improvement Techniques

1996 Sloan Management Review, Fall, 1996

Field, Tom

When bad things happen to good projects

1997 Computing SA, November 1997

Furlong, Steve

Saving a Project in Trouble

1997 Software Developmenr, October 1997

Hamel, G.

Strategy Innovation and the Quest for Value

1998 Sloan Management Review, Winter 1998

Handy, Charles

Trust and the Virtual Organisation

1995 Harvard Business Review, May-June 1995

Highsmith, Jim

Order for Free

1998 Software Development, March 1998

Iansiti, M. & West, J.

Technology Integration: Turning Great Research into Great Products

1997 Harvard Business Review, May-June 1997

Keuffel, Warren

Eclectic Requirements Gathering

1998 Software Development, March 1998

Keuffel, Warren

Fictitious Deadlines

1998 Software Development, January 1998

Kim, W.C. & Mauborgne, R.

Fair Process: Managing in the Knowledge Economy

1997 Harvard Business Review, July-August, 1997

Kleiner, A. & Roth, G.

How to Make Experience Your Company's Best Teacher

1997 Harvard Business Review, September-October 1997

Krackhardt, D. & Hanson, J.R.

Informal Networks: The Company behind the chart

1993 Harvard Business Review, July-August, 1993

Lacity, M. C., Willcocks, L. P. & Feeny, D. F.

IT Outsourcing: Maximize Flexibility and Control

1995 Harvard Business Review, May-June 1995

Leavitt, H. J. & Lipman-Blumen, J.

Hot Groups

1995 Harvard Business Review, July-August 1995

Leffingwell, Dean

Understanding User Needs

1997 Software Development, August 1997

Leonard, D. & Straus, S

Putting Your Company's Whole Brain To Work

1997 Harvard Business Review, July-August, 1997

Lester, R., Piore, M. & Malek, K.

Interpretive Management: What General Managers can Learn from Design

1998 Harvard Business Review, March-April 1998

Lieitka, J.M., Haskins, M.E., Rosenblum, J.W. & Weber, J.

The Generative Cycle: Linking Knowledge and Relationships

1977 Sloan Management Review, Fall 1977

Lipton, Mark

Demystifying the Development of an Organisational Vision

1996 Sloan Management Review, Summer 1996

Loomis, Mary

Managing Collaborations

1998 Software Development, January 1998

MacMillan, Michael

Finding Common Ground

1998 Computing SA, 8 June, 1998

Malone, Thomas W.

Is Empowerment Just a Fad? Control, Decision Making and IT?

1997 Sloan Management Review, Winter 1997

Markus, M.L. & Benjamin, R.I.

The Magic Bullet Theory in IT-Enabled Transformation

1997 Sloan Management Review, Winter 1997

Mishra, K.E., Spreitzer, G.M. & Mishra, A.K.

Preserving Employee Morale during Downsizing

1998 Sloan Management Review, Winter 1998

Nicholson, Nigel

How Hardwired Is Human Behaviour?

1998 Harvard Business Review, July - August 1998

Nolan, Richard et al

Connectivity and Control in the Year 2000 and Beyond

1998 Harvard Business Review, July-August 1998

Orlikowski, W.J & Hofman, J.D.

An Improvisational Model for Change Manegement: The Case of Groupware Technologies

1997 Sloan Management Review, Winter 1997

Prokesch, Steven E.

Unleashing the Power of Learning: An Interview with British Pretroleum's John Brown

1997 Harvard Business Review, September-October 1997

Quinn, J.B., Baruch, J.J., & Zein K. A.

Software-Based Innovation

1996 Sloan Management Review, Summer 1996

Racko, Roland

Debugging Meetings

1997 Software Development, September 1997

Reimus, Byron

The IT System That Couldn't Deliver

1997 Harvard Business Review, May-June 1997

Ross, J.W., Beath, C. M. & Goodhue, D.L.

Develop Long-Term Competitiveness through IT Assets

1996 Sloan Management Review, Fall 1996

Schein, Edgar H.

Three Cultures of Management: The Key to Organisational Learning

1996 Sloan Management Review, Fall 1996

Schrage, Michael

The Real Problem With Computers

1997 Harvard Business Review, September - October, 1997

Shapiro, Eileen

Managing in the Age of Gurus

1997 Harvard Business Review, March-April 1997

Shaw, G., Brown, R. & Bromily, P.

Strategic Stories: How 3M is Rewriting Business Planning

1998 Harvard Business Review, May-June 1998

Surveyer, Jacques

Developers' Angst and Anger

1998 Software Development, April 1998

Tabrizi, B. & Walleigh, R.

Defining Next Generation Products: An inside look

1997 Harvard Business Review, November-December, 1997

Tannen, Deborah

The Power of Talk: Who gets heard and why

1996 Harvard Business Review, May-June 1996

Ulrich, Dave

Intellectual Capital = Competence X Commitment

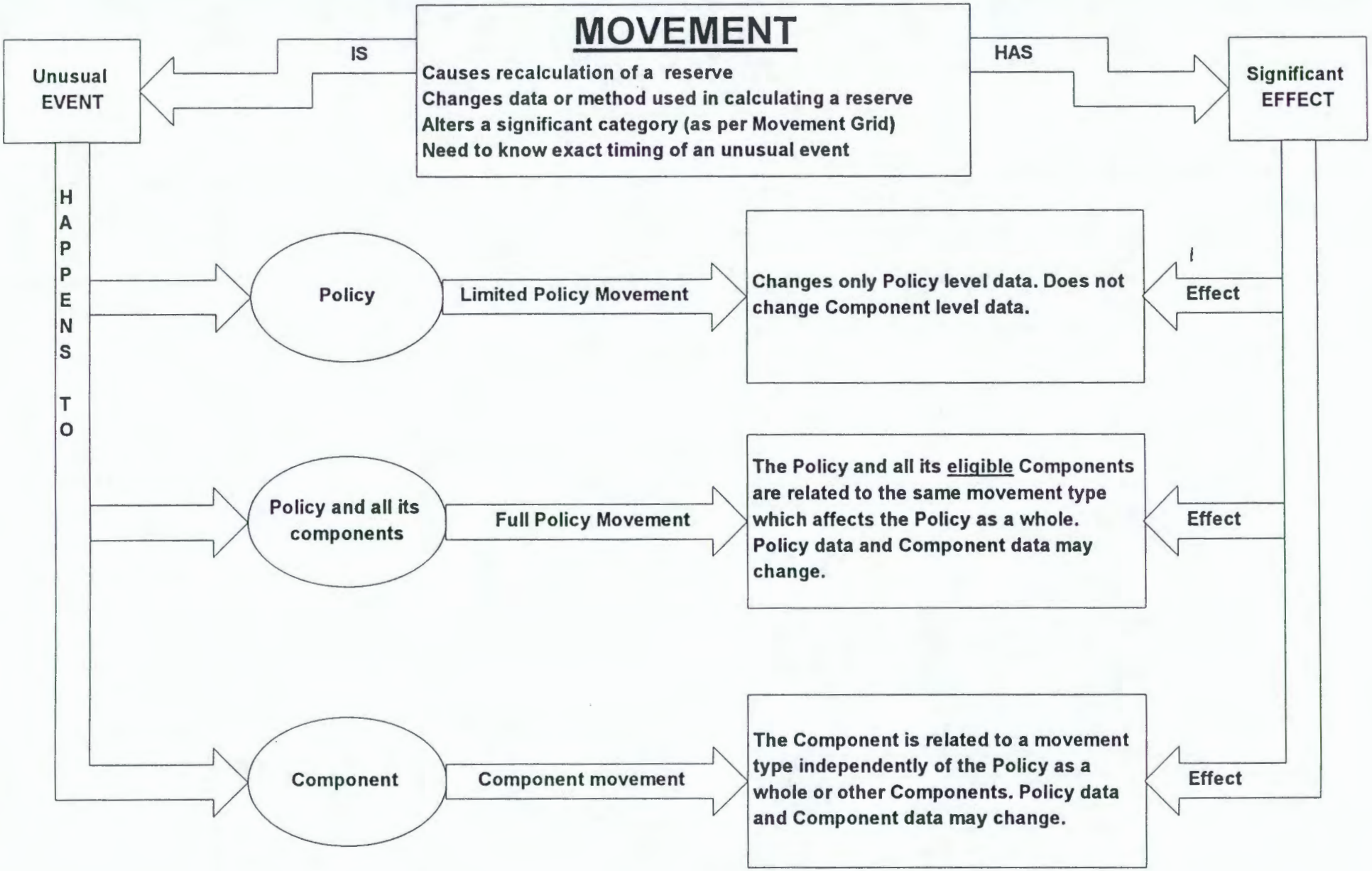
1998 Sloan Management Review, Winter 1998

Venkatraman, N.

Beyond Outsourcing: Managing IT Resources as a Value Center

1997 Sloan Management Review, Spring, 1997

APPENDIX A


MOVEMENT
Causes recalculation of a reserve
Changes data or method used in calculating a reserve
Alters a significant category (as per Movement Grid)
Need to know exact timing of an unusual event
Unusual EVENT
IS
HAS
Significant EFFECT
HAPPENS TO
Policy
Limited Policy Movement
Changes only Policy level data. Does not change Component level data.
Effect
Policy and all its components
Full Policy Movement
The Policy and all its eligible Components are related to the same movement type which affects the Policy as a whole. Policy data and Component data may change.
Effect
Component
Component movement
The Component is related to a movement type independently of the Policy as a whole or other Components. Policy data and Component data may change.
Effect

A1

A2

Out of Force Movements

Movement ID	Movement Name	Movement Scope	Movement Status Change	Movement Type	VANTAGE Transaction	VANTAGE Policy Status	VANTAGE Comp Status
BCN	Benefit Cancellation	Component	INF - OOF	Termination	GLD		Terminated
CDC	Capital Disability claim	Full Policy	INF - OOF	Termination	GH	Terminated	
DIS	Component Disability claim	Component	INF - OOF	Termination	GH		Terminated
DRD	Component Dread disease claim	Component	INF - OOF	Termination	GH(A)		Terminated
EXP	Component Expiry	Component	INF - OOF	Termination	FNT		Terminated
CLP	Component Lapse	Component	INF - OOF	Termination	TL	Reduced Paid-Up	Lapsed
MAT	Component Maturity	Component	INF - OOF	Termination	FNM		Matured
CSR	Component Surrender	Component	INF - OOF	Termination	TA / TB		Surrendered
CTR	Component Termination	Component	INF - OOF	Termination	NFX		Terminated
ERT	Early retirement	Full Policy	INF - OOF	Termination	TA(=SV)	Surrender	
ESR	Equity surrender	Component	INF - OOF	Termination	FNS - TL		Lapsed
LSR	Loan Surrender	Full Policy	INF - OOF	Termination	TA(PAY)NE	Surrender	
NRT	Normal Retirement	Full Policy	INF - OOF	Termination	FNM	Matured	
NTU	Not taken up	Full Policy	INF - OOF	Termination	NFX(N)	Not Taken	
CFR	Cancelled with full refund	Full Policy	INF - OOF	Termination	QAC	Cancelled	
PDT	Policy Death claim	Full Policy	INF - OOF	Termination	PA	Death	
PDC	Policy Dread disease claim	Full Policy	INF - OOF	Termination	GH(A)	Terminated	
PEX	Policy Expiry	Full Policy	INF - OOF	Termination	FNE	Expired	
PLA	Policy Lapse	Full Policy	INF - OOF	Termination	TL	Lapsed	
PMT	Policy Maturity	Full Policy	INF - OOF	Termination	FNM(M)	Matured	
PSU	Policy Surrender	Full Policy	INF - OOF	Termination	TA	Surrendered, Lapsed	
RDC	Repudiation of death claim	Component	OOF - OOF	Alteration	PX	Rescinded	
TPW	Termination of Premium Waiver	Component	OOF - OOF	Alteration	DD / GH		Terminated
WAD	Waiver of premium claim - disability	Component	INF - OOF	Termination	GH		Terminated
ACT Movements which have not been Implemented on VANTAGE							
TRM	Conversion termination	Full Policy	INF - OOF	Termination	Not exist on VANTAGE		
CDT	Component Death claim	Component	INF - OOF	Termination	Future		
WAP	Waiver of premium claim - death	Component	INF - OOF	Termination	Future		

Revivals

Movement ID	Movement Name	Movement Scope	Movement Status Change	Movement Type	VANTAGE Transaction	VANTAGE Policy Status	
XLP	Revival ex Lapse	Full Policy	OOF - INF	Revival	NFXR	Reinstated	

15 July 1996 — QPM & Lapse requirements

KEY :
C = Chris; J = Jeremy;
N = Janina; M = Marie;
T = Trevor; X = All

Inforce Movements

Movement ID	Movement Name	Movement Scope	Movement Status Change	Movement Type	VANTAGE Transaction	VANTAGE Policy Status	VANTAGE Comp Status
NBS	New Business	Full Policy	0 - INF	Inception	PRC	Await Init Pay	
FPP	First Premium payment	Full Policy	INF - INF	Alteration	NA	Active	
CON	ALIS Conversion	Full Policy	INF - INF	Inception	PRC	Active	
VLI	Voluntary increase	Component	INF - INF	Alteration	GLI,GC,GG,GF,GX		Active
UAD	Inflation Update Add	Component	INF - INF	Alteration	GN		Active
IFU	Inflation Update (incl. Compulsory update)	Component	INF - INF	Alteration	GLC/MP		Active
IFC	Inflation Update Cancellation	Component	INF - INF	Alteration	IF		Active
IFR	Inflation Update Refusal	Component	INF - INF	Alteration	IF		Active
SPI	Single premium injection	Component	INF - INF	Alteration	MA		Active
WTH	Cash Withdrawal	Component	INF - INF	Alteration	TW		Active
APU	Component Auto Paid Up	Component	INF - INF	Alteration	TLN	Reduced Paid-Up	Paid-Up
RPU	Component Requested Paid Up	Component	INF - INF	Alteration	TLN	Reduced Paid-Up	Paid-Up
HIV	Failure to provide negative HIV test	Component	INF - INF	Alteration	HT		Active
FTR	Fund Transfer	Component	INF - INF	Alteration	OE		Active
LPT	Limited-Premium Term expired	Component	INF - INF	Alteration	FNP		Paid-Up
DFM	Maturity Deferred	Component	INF - INF	Alteration	GO		Open-ended
OPT	Option take up	Component	INF - INF	Alteration	GW		Active
PLP	Part Lapse	Component	INF - INF	Alteration	PRM		RPU
PSU	Part Surrender	Component	INF - INF	Alteration	TW/PRM		RPU
XPU	Reinstatement ex Reduced Paid Up	Component	INF - INF	Alteration	NFXR / RAX ?		Active
PAY	Claim Payment	Component	INF - INF	Alteration	PX		Active
REP	Repudiation of Claim	Component	INF - INF	Alteration	PX		Active
PDS	Partial Disability	Component	INF - INF	Alteration	GH		Active
PDX	Partial disability (Xenon)	Component	INF - INF	Alteration	GH		Active
PDD	Partial Dread disease claim	Component	INF - INF	Alteration	GH		Active
DCO	Disability claim on other component	Component	INF - INF	Alteration	GH		Active
DDO	Dread disease claim on other component	Component	INF - INF	Alteration	GH		Active
LCD	Limited Capital Disability claim	Component	INF - INF	Alteration	GH		Active
PXO	Partial xenon claim on other component	Component	INF - INF	Alteration	GH		Active
PRG	Pregnancy suspension claim	Component	INF - INF	Alteration	GH		Active
TPS	Termination of Premium Suspension Claim	Component	INF - INF	Alteration	GH		Active
UMS	Unemployment suspension claim	Component	INF - INF	Alteration	GH		Active
WAV	Waiver / Suspension of claims on other comp	Component	INF - INF	Alteration	GH		Active
ALT	General Alteration	Full Policy	INF - INF	Alteration	PRC		Active
New ACT Movements							
CDX	Component Disability claim (Xenon)	Component	INF - INF	Alteration	GH		Active
	Death of a Life Assured				Future		
	Addition of a Life Assured				Future		
	Removal of a Life Assured				Future		
	Change in Smoker Status				Future		
ACT Movements which may be invalid specifications							
	Policy Auto Paid Up	Full Policy	INF - INF	Alteration			
	Policy Requested Paid Up	Full Policy	INF - INF	Alteration			
	Reinstatement ex Policy Auto Paid Up	Full Policy	INF - INF	Alteration			
	Reinstatement ex Policy Requested Paid Up	Full Policy	INF - INF	Alteration			
ACT Movements which have not been implemented on VANTAGE							
SPH	Stepping PHI claim	Component	INF - INF	Alteration	Future		
ARA	Actuarial Review Alteration	Component	INF - INF	Alteration	Not identifiable		
DJL	Change by death on Joint lives	Component	INF - INF	Alteration	Not Implemented		
LEG	Change in Legal class	Limited Policy	INF - INF	Alteration	Not Implemented		
TAX	Change in Tax category	Limited Policy	INF - INF	Alteration	Not Implemented		
TER	Change in Territory	Limited Policy	INF - INF	Alteration	Not Implemented		
DCC	Death claim with continuance	Component	INF - INF	Alteration	Not Implemented		
PHI	PHI claim	Component	INF - INF	Claim ??	On RPS system		

Out of Force Movements

Movement ID	Movement Name	Movement Scope	Movement Status Change	Movement Type	VANTAGE Transaction	VANTAGE Policy Status	VANTAGE Comp Status
BCN	Benefit Cancellation	Component	INF - OOF	Termination	GLD		Terminated
CDC	Capital Disability claim	Full Policy	INF - OOF	Termination	GH	Terminated	
DIS	Component Disability claim	Component	INF - OOF	Termination	GH		Terminated
DRD	Component Dread disease claim	Component	INF - OOF	Termination	GH(A)		Terminated
EXP	Component Expiry	Component	INF - OOF	Termination	FNT		Terminated
CLP	Component Lapse	Component	INF - OOF	Termination	TL	Reduced Paid-Up	Lapsed
MAT	Component Maturity	Component	INF - OOF	Termination	FNM		Matured
CSR	Component Surrender	Component	INF - OOF	Termination	TA / TB		Surrendered
CTR	Component Termination	Component	INF - OOF	Termination	NFX		Terminated
ERT	Early retirement	Full Policy	INF - OOF	Termination	TA(=SV)	Surrender	
ESR	Equity surrender	Component	INF - OOF	Termination	FNS - TL		Lapsed
LSR	Loan Surrender	Full Policy	INF - OOF	Termination	TA(PAY)NE	Surrender	
NRT	Normal Retirement	Full Policy	INF - OOF	Termination	FNM	Matured	
NTU	Not taken up	Full Policy	INF - OOF	Termination	NFX(N)	Not Taken	
CFR	Cancelled with full refund	Full Policy	INF - OOF	Termination	QAC	Cancelled	
PDT	Policy Death claim	Full Policy	INF - OOF	Termination	PA	Death	
PDC	Policy Dread disease claim	Full Policy	INF - OOF	Termination	GH(A)	Terminated	
PEX	Policy Expiry	Full Policy	INF - OOF	Termination	FNE	Expired	
PLA	Policy Lapse	Full Policy	INF - OOF	Termination	TL	Lapsed	
PMT	Policy Maturity	Full Policy	INF - OOF	Termination	FNM(M)	Matured	
PSU	Policy Surrender	Full Policy	INF - OOF	Termination	TA	Surrendered, Lapsed	
RDC	Repudiation of death claim	Component	OOF - OOF	Alteration	PX	Rescinded	
TPW	Termination of Premium Waiver	Component	OOF - OOF	Alteration	DD / GH		Terminated
WAD	Waiver of premium claim - disability	Component	INF - OOF	Termination	GH		Terminated
ACT Movements which have not been implemented on VANTAGE							
TRM	Conversion termination	Full Policy	INF - OOF	Termination	Not exist on VANTAGE		
CDT	Component Death claim	Component	INF - OOF	Termination	Future		
WAP	Waiver of premium claim - death	Component	INF - OOF	Termination	Future		

Revivals

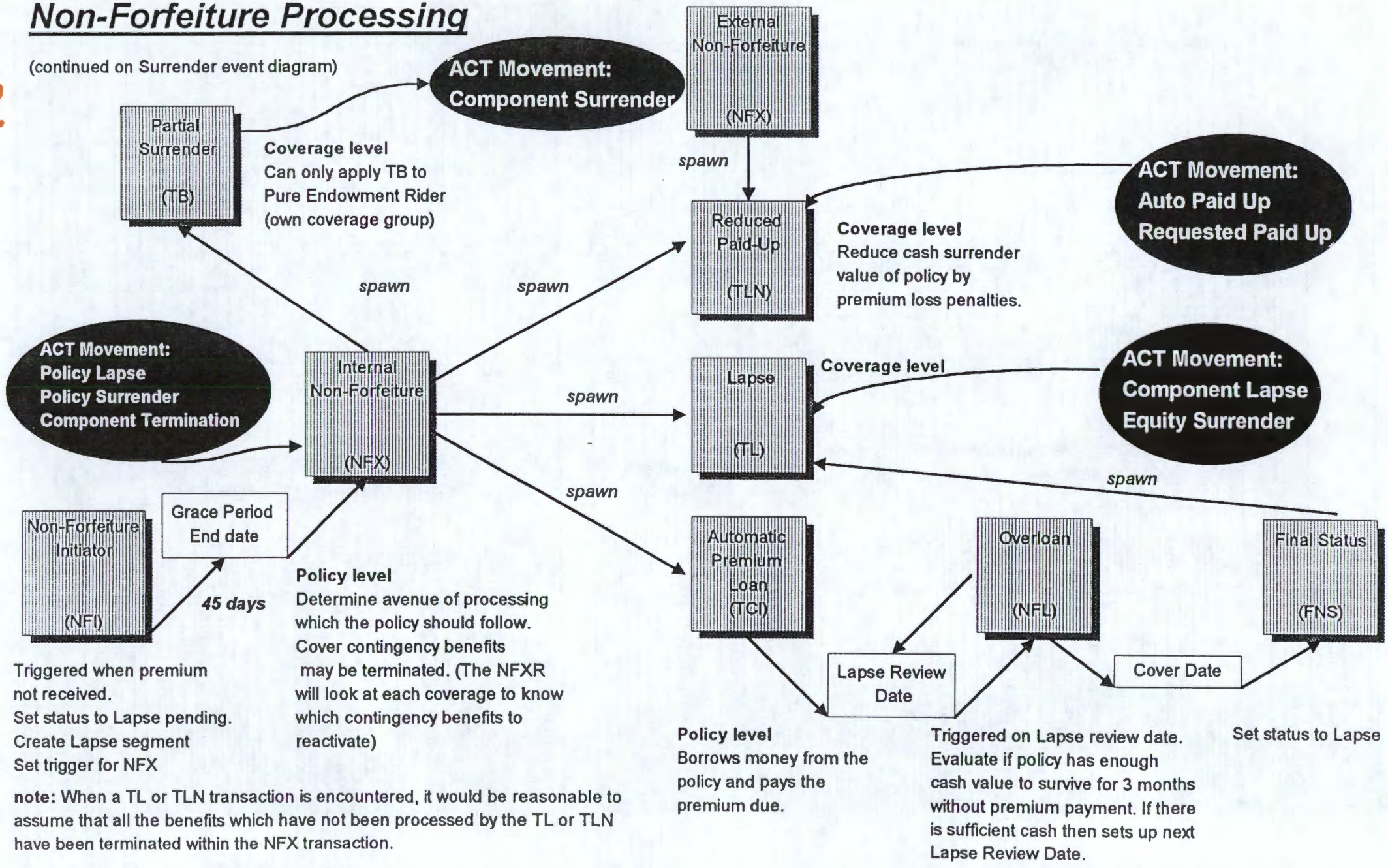
Movement ID	Movement Name	Movement Scope	Movement Status Change	Movement Type	VANTAGE Transaction	VANTAGE Policy Status	
XLP	Revival ex Lapse	Full Policy	OOF - INF	Revival	NFXR	Reinstated	

A3

Irregular Unit Linked Transaction Processing:
Non-Forfeiture Processing

A4

(continued on Surrender event diagram)



note: When a TL or TLN transaction is encountered, it would be reasonable to assume that all the benefits which have not been processed by the TL or TLN have been terminated within the NFX transaction.

Regular Unit Linked Transaction Processing:

Note: For all policies which will have been converted from ALIS, the Commencement day is on the first of the month.

ACT Movement: Inflation Update

Month

Day 1

Cost of Living Adjustment (GLC)

The Cost of Living adjustment is processed annually on the same Day as the Policy Commencement Day.

Payment Remittance (NA)

For the time being the Premiums received will be processed on the first of each month, as this is when the unit prices are available.

Cost of Insurance & Monthly Expense Charges (XA)

Reassurance (RAU)

The Cost of Insurance and Reassurance is processed each month on the same Day as the Policy Commencement Day.

The XA transaction on VANTAGE results in two investment cashflows on ACT.

Day 15

Automatic Premium loan payment (NAI)

The Automatic Premium Loan premium payment will occur around the 15th of each month as this is 45 days after the 1st of the previous month. This means that the policy holder did not pay his premium on the 1st of the previous month. 45 days is the Grace period before the Automatic Premium Loan is activated.

Everyday

Run Control (AA)

The Run control transaction enters the control data necessary to successfully operate the administration cycle. This is where the Unit Prices are updated on the VANTAGE database.
(The prices are updated everyday, even if they haven't changed.)

Annual Dividend Declaration (WF)

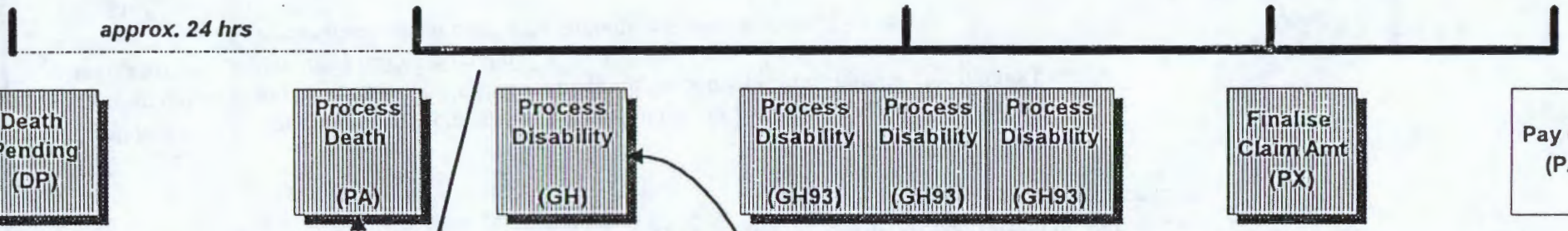
The Annual Dividend declaration will be run at the Financial Year end, when it is requested by Corporate Actuarial.

AS

A6

Irregular Unit Linked Transaction Processing:
Death / Disability claims

The time delays between the occurences of these transactions are not predefined.



This transaction is optional and will not be used often.

First notification that a death has occured. Used to stop billing i.e prevent unnecessary activity.

There is an option on this transaction (currently not used) to transfer money to another fund. We may want to use this for UAL linked business to transfer the policyholder's money from a volatile to a fixed interest fund.

ACT Movement:
Policy Death Claim

Policy (insured) level transaction
Set up Claim segment
Calculate Death / Disability Amt
Set policy status to terminated
Terminate benefits for Disability
Perform Accounting movements

Admit Waiver of premium benefit for Disability (GH).
Set up the date for when the Waiver benefit must cease.

Coverage level

ACT Movement:
Capital Disability
Limited Capital Disability claim
Component Dread disease claim
Policy Dread disease claim
Partial Dread Disease claim
Dread disease Claim on other component
Partial Disability (Xenon)
Partial Xenon Claim on other component
Waiver of premium claim - disability
Unemployment Suspension claim
Pregnancy Suspension claim
Termination of Premium Waiver
Termination of Suspension claim

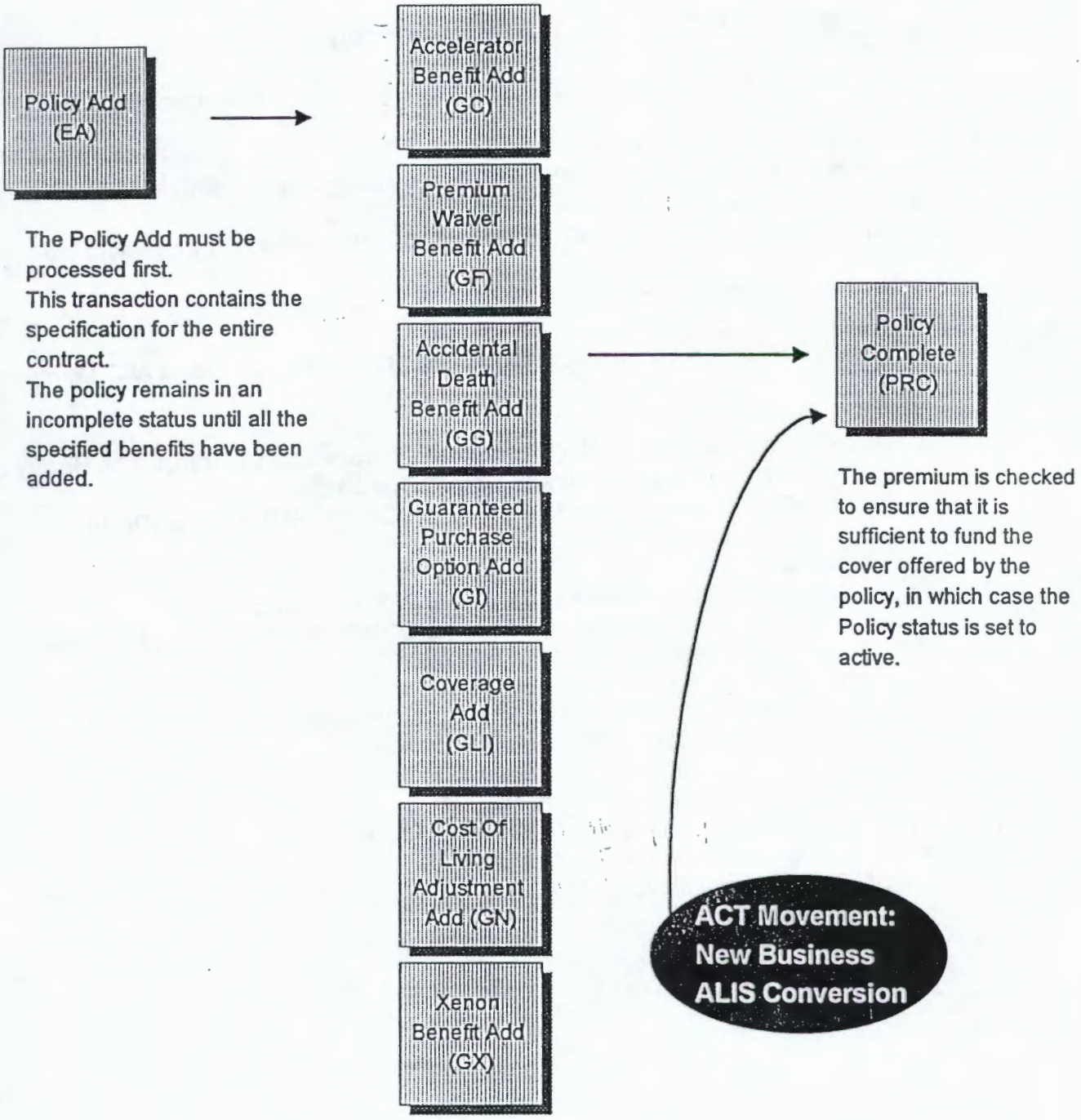
Policy level transaction
Adjust Claim Amount
= final liability
(repudiate / ex-gratia)
Perform Accounting movements
Flag Claim Portion as paid

ACT Movement:
Repudiation of Claim
Ex-Gratia Claim Payment

Money paid out to Policyholder or transferred to SUTCO / Annuities

Irregular Unit Linked Transaction Processing: New Business / ALIS Conversion

These transaction are executed immendiately after each other, in any order.



The Policy Add must be processed first.
This transaction contains the specification for the entire contract.
The policy remains in an incomplete status until all the specified benefits have been added.

The premium is checked to ensure that it is sufficient to fund the cover offered by the policy, in which case the Policy status is set to active.

If these transactions had been used to convert a policy from ALIS to the Unit Linked database then they would be marked by an ALIS conversion indicator.

A7

Non-Forfeiture Processing in the Movement Interface

For Each Lapse (TL) Transaction which was spawned by a Final status (FNS) Transaction
(The TL transaction(s) are immediately succeeded by a FNS transaction on Transaction History)
Set the ACT Component Movement to **EQUITY SURRENDER** by creating ACT Movement table

For Each Non-Forfeiture (NFX) Transaction in Transaction History
If VANTAGE Policy-Status = Not Taken
Set Full Policy Movement to **NOT TAKEN UP** by creating ACT Movement table

If VANTAGE Policy-Status = Lapsed
Set Full Policy Movement to **POLICY LAPSE** by creating ACT Movement table
If VANTAGE Policy-Status = Surrender
Set Full Policy Movement to **POLICY SURRENDER** by creating ACT Movement table

If VANTAGE Policy-Status = RPU
For each Lapse (TL) (coverage level) Transaction
Set ACT Component Movement to **COMPONENT LAPSE** by creating ACT Movement table

For Each Reduced Paid-Up (TLN) (coverage level) Transaction in Transaction History
If the TLN Transaction is followed by a TB Transaction
Set ACT Component Movement to **COMPONENT SURRENDER** by creating ACT Movement table
otherwise
If the TLN was spawned from an External NFX Transaction
Set ACT Component Movement to **COMPONENT REQUESTED PAID UP** by creating ACT Movement table
otherwise
Set ACT Component Movement to **COMPONENT AUTO PAID UP** by creating ACT Movement table

For Each ACT Component which has not had a
VANTAGE TL transaction nor a TLN transaction processed and
the ACT Component is Active and not Paid Up
Set ACT Component Movement to **COMPONENT TERMINATION** by creating ACT Movement table

A8

Reversal of
Non-Forfeiture Processing in the ACT Movement Interface

For Each Policy
Read VANTAGE Transaction History from the start of the ACT Movement window

For Each TLR Transaction that is preceded by a Final status Reversal (FNSR) Transaction
Reverse ACT Component movement(s) = EQUITY SURRENDER

If there is a Non-Forfeiture Reversal (NRXR) Transaction in Transaction History
IF Prior-Status (NFXR transaction) is Not taken
Reverse ACT Full Policy movement = NOT TAKEN UP

IF Prior-Status (NFXR transaction) is Lapsed
Reverse ACT Full Policy movement = POLICY LAPSE

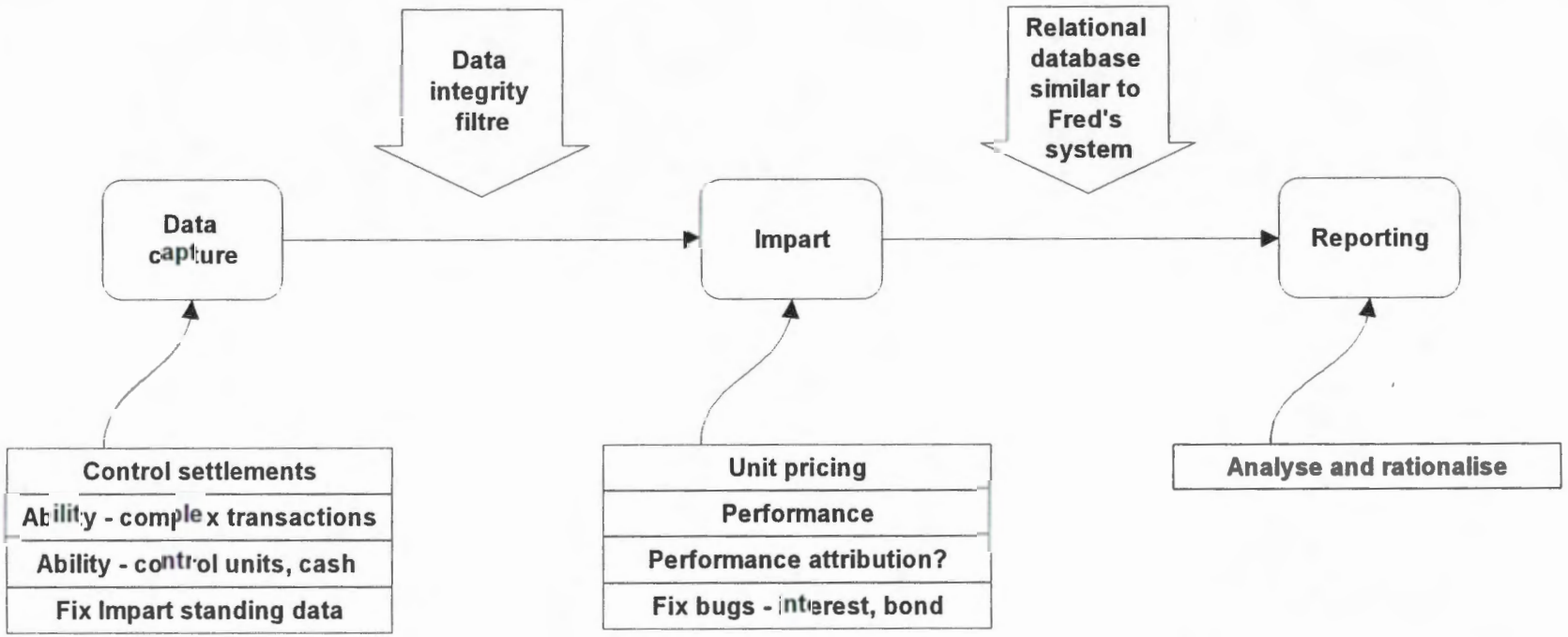
IF Prior-Status (on NFXR transaction) is Reduced Paid Up
For each Lapse Reversal (TLR) (coverage level) Transaction
Reverse ACT Component movement(s) = COMPONENT LAPSE

For Each Reduced Paid-Up Reversal (TLNR) (coverage level) Transaction in Transaction History
If the TLNR Transaction is preceded by a TBR Transaction
Reverse ACT Component movement(s) = COMPONENT SURRENDER
otherwise
If the TLNR was spawned from an External NFXR Transaction
Reverse ACT Component movement(s) = REQUESTED PAID UP
otherwise
Reverse ACT Component movement(s) = AUTO PAID UP

Reverse ACT Component movement(s) = COMPONENT TERMINATION

APPENDIX B

MINIMUM DELIVERABLE FOR COPING WITH CORE & SATELITE?



Flesh out:
1. Who is allocated to each task
2. Critical dependencies for each task
3. Project management routine
4. Managing the communication channels
5. Target a skills growth plan

ARGUMENT

MODEL BUILDING

VOCABULARY

Collaborative project:
"ACTS of SHARED CREATION"
Derive 3 domains & 3 subsystems
Describe Table 1 structure, category types
Chapter 1

3 DOMAINS OF COLLABORATIVE PROJECTS

ACTING (Acts)	THINKING (Creation)	CONVERSING (Shared)

3 SUBSYSTEMS:
Language (Human activity)
Metaphor (Conceptual)
Conversation (Interpretative)

18 category types

Step 1
Categories for the design of collaborative projects
Develop theoretical and philosophical positions for each domain and category type.
Derive a set of distinctions.
Develop a set of categories and insights to aid the designer.
Justify 12 influencers.
Chapter 3

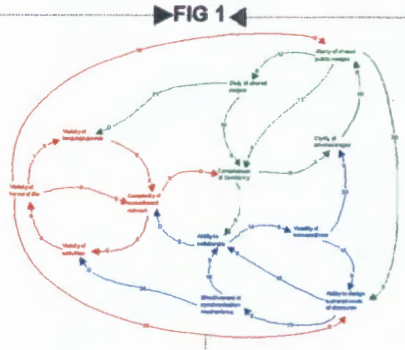
TABLE 1

CATEGORY TYPE	ACTING	THINKING	CONVERSING
Domain			
Conversation type			
Process			
Intervention			
Influencer	red	green	blue

12 INFLUENCERS:
Activities
Forms of life
Commitment networks
Language games
Private images
Shared public images
Community
Shared purpose
Viable conversation
Shared mode of discourse
Collaboration network
Synchronization mechanisms

Step 2
Causal influences of collaborative projects
Derive the social process underpinning collaborative projects and represent it as a system of causal influences using influencers identified in Table 1.
Identify the common structure of the 3 subsystems.
Derive measurabilty for 12 influencers.
Introduce 14 Keywords.
Chapter 4

FIG 1



14 KEYWORDS:
Socializing
Worldmaking
Forming
Designing
Governing
Legitimizing
Normalizing
Performing
Trusting
Representing
Transforming
Expressing
Explicating
Co-ordinating

12 MEASURABLE INFLUENCERS:
Variety of activities
Variety of forms of life
Complexity of commitment network
Variety of language games
Clarity of private images
Clarity of shared public images
Cohesiveness of community
Clarity of shared purpose
Viability of conversations
Ability to design a shared mode of discourse
Ability to collaborate
Effectiveness of synchronization mechanisms

Step 3
Appreciative systems framework
Justify collaborative projects as appreciative systems. Link the appreciative systems framework to the social process and causal influences developed via Fig 1.
Overlay the common structure of the 3 subsystems w ith the appreciative systems framework.
Justify 5 intra-subsystem movements.
Justify 4 influencer types.
Chapter 5

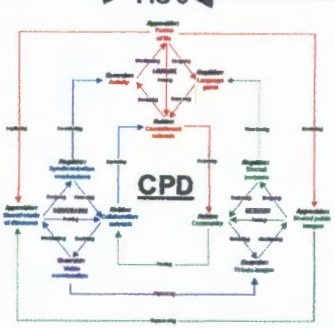
FIG 2

4 INFLUENCER TYPES:
Generator
Appreciator
Relater
Regulator

5 INTRA-SUBSYSTEM MOVEMENTS:
Socializing
Worldmaking
Forming
Designing
Governing

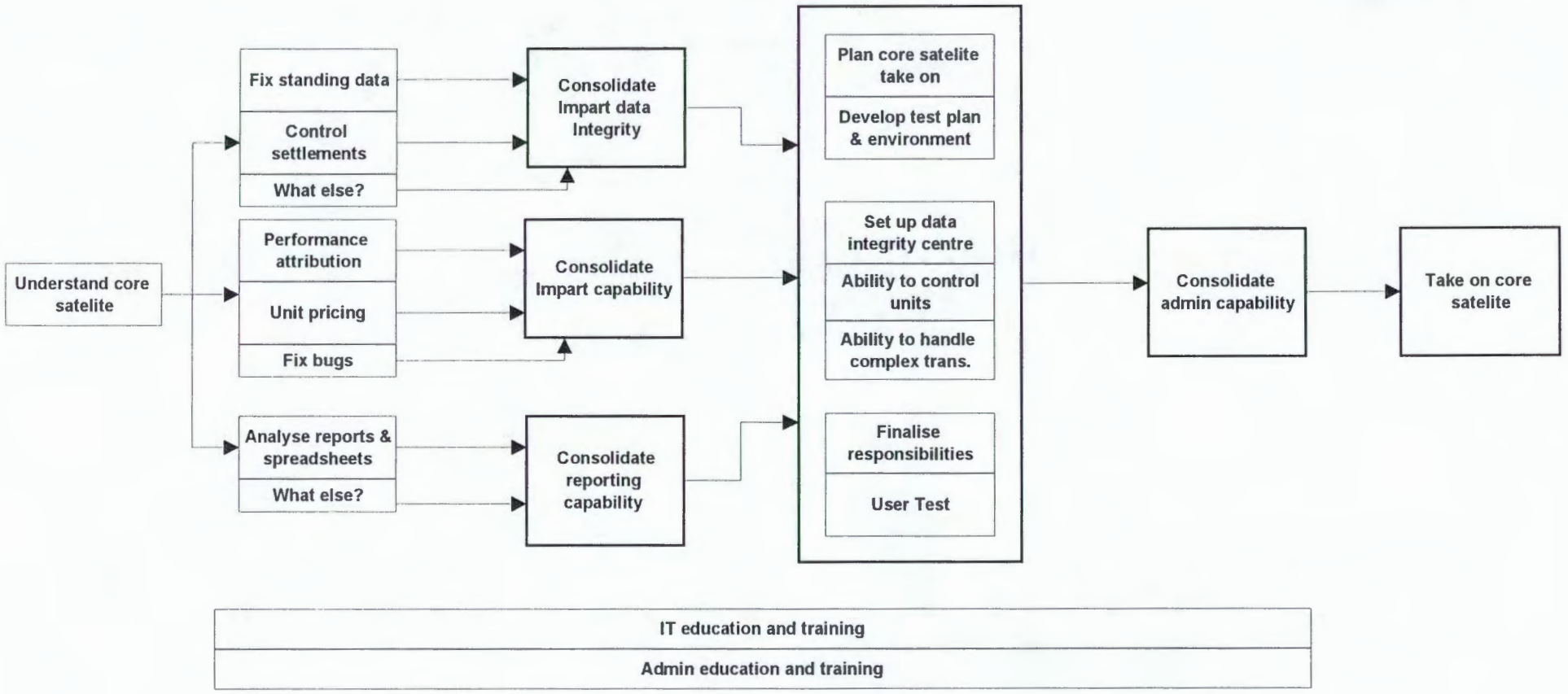
Step 4
Model for collaborative project design
Extend the appreciative systems framework by overlaying it onto the full set of causal influences represented in Fig 1.
Justify 9 inter-subsystem movements.
Chapter 6

FIG 3



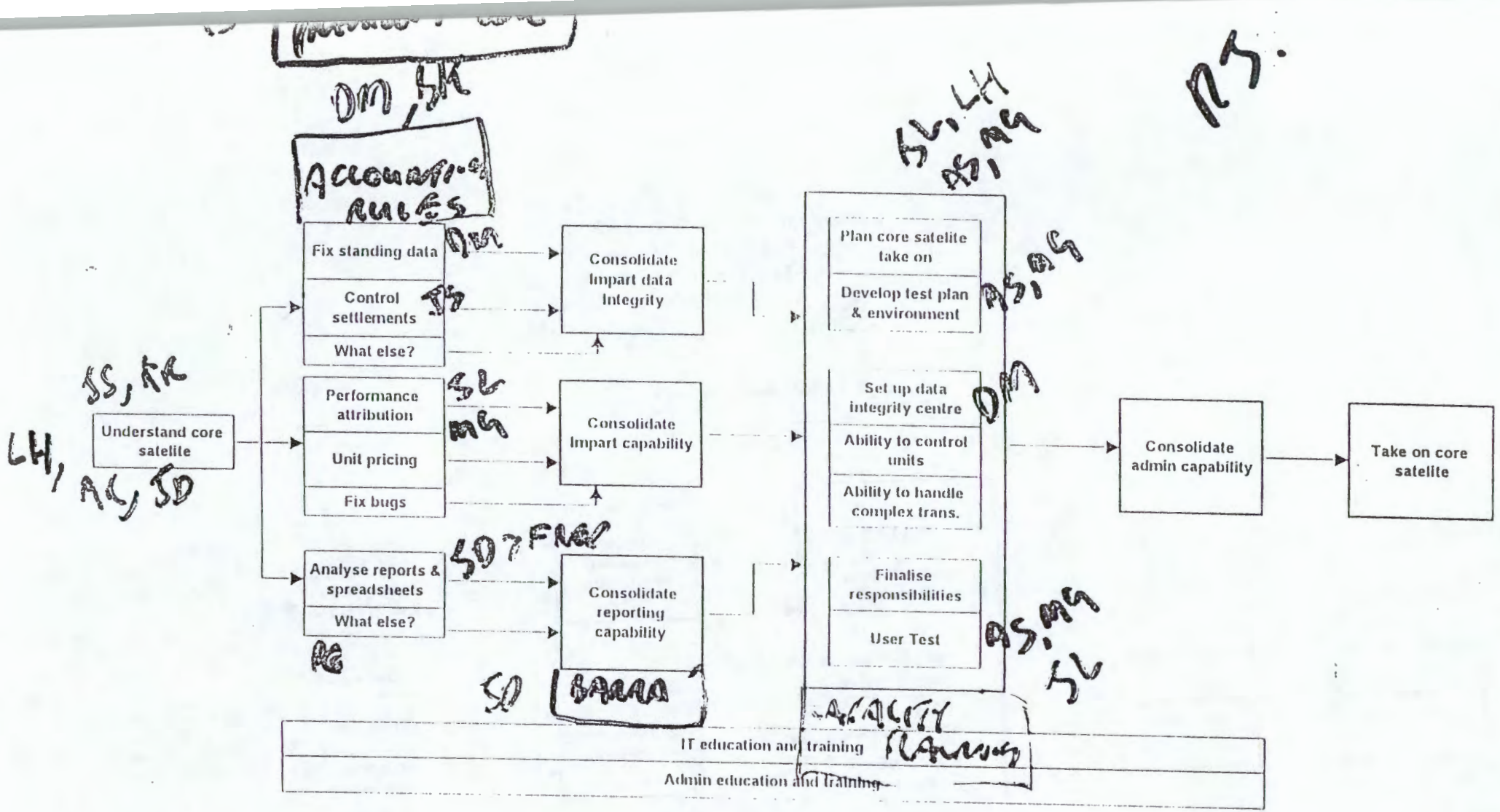
9 INTER-SUBSYSTEM MOVEMENTS:
Legitimizing
Normalizing
Performing
Trusting
Representing
Transforming
Expressing
Explicating
Co-ordinating

Schema 1 - Outline of thesis structure



Flesh out:

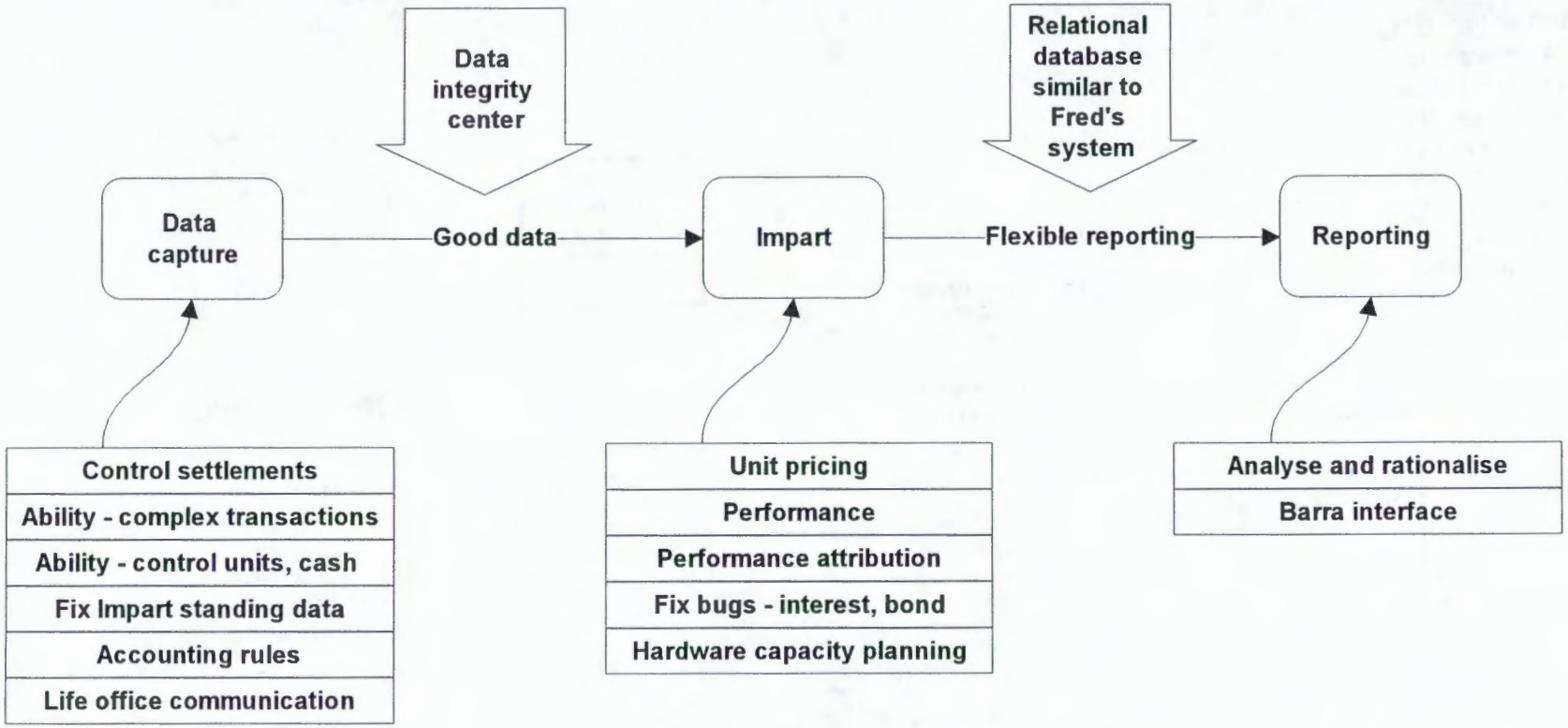
1. Who is allocated to each task
2. Critical dependencies for each task
3. Project management routine
4. Managing the communication channels
5. Target a skills growth plan



Flesh out:

1. Who is allocated to each task
2. Critical dependencies for each task
3. Project management routine
4. Managing the communication channels
5. Target a skills growth plan

MINIMUM DELIVERABLE FOR COPING WITH CORE & SATELITE



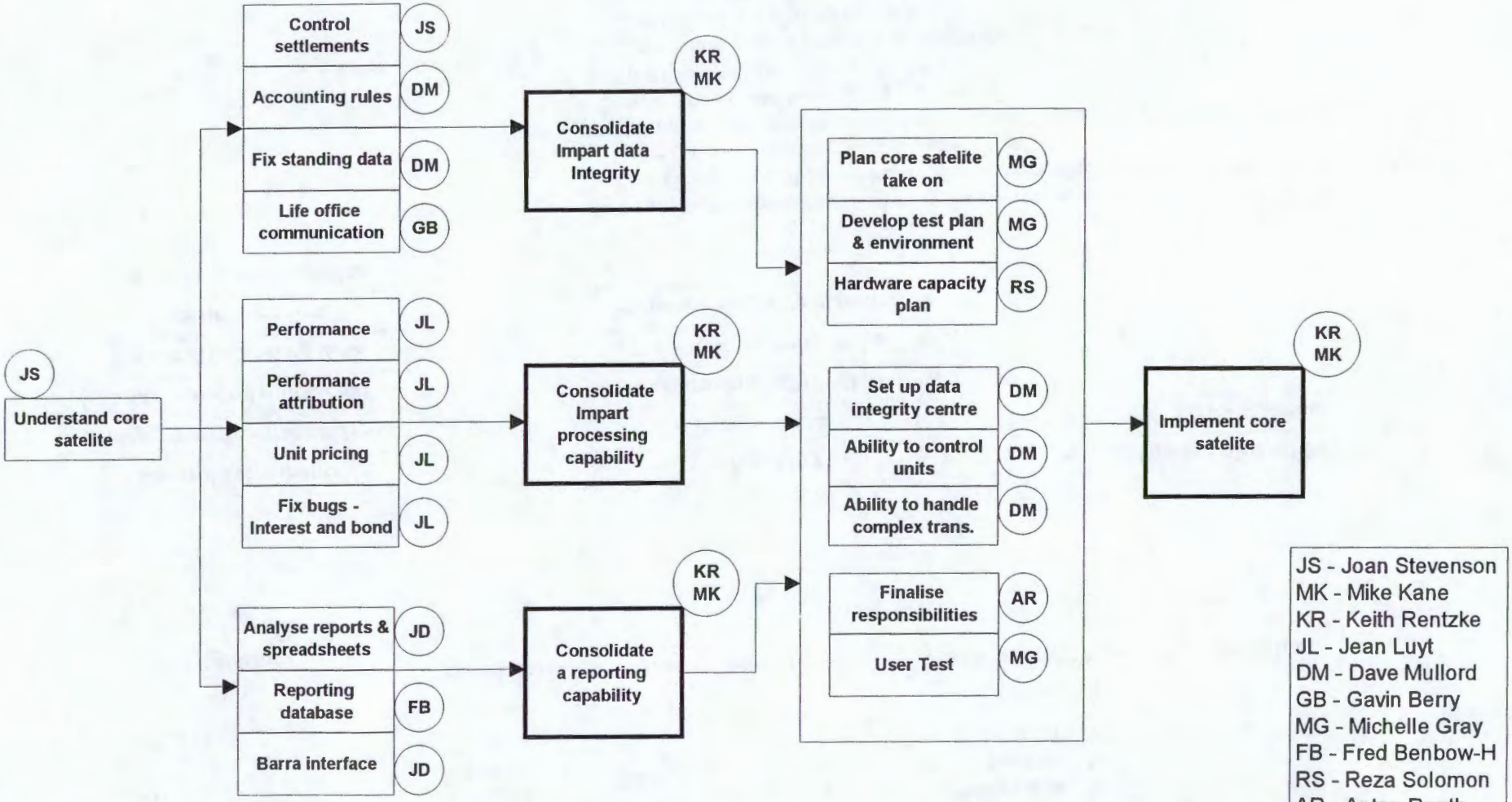
The minimum deliverable has two main components:
1. Stabalise Impart, Improve admin procedures, Ensure data integrity ("the old problem")
2. Cater for core satelite ("the new challenge")

Each of the above blocks is needed to support the minimum deliverable and provide data integrity. There are four major projects:
1. Consolidate Impart data integrity
2. Consolidate Impart processing capability
3. Consolidate a reporting capability
4. Implement core satelite

There is a general feeling that it is possible to meet this deliverable by 1st May, 1997.

49

PROJECTS, SUB-PROJECTS AND PROJECT DRIVERS



RS

Project resources

This represents peoples roles *as specifically assigned during the planning session* . This list will have to be developed further as the project's scope and plans are planned in finer detail.

Consolidate Impart data integity

Control settlements	Joan Stevenson (driver)
Accounting rules	Dave Mullord (driver), Butana Khosa
Fix standing data	Dave Mullord (driver)
Life office communication	Gavin Berry (driver), Jean Luyt

Consolidate Impart Processing

Performance	Jean Luyt (driver), Michelle Gray
Performance attribution	Jean Luyt (driver), Michelle Gray
Unit pricing	Jean Luyt (driver), Michelle Gray
Fix bugs - Interest, Bond	Jean Luyt (driver), Michelle Gray

Consolidate a Reporting Capability

Analyse spreadsheets and reports	Julian Day (driver), Geoff Chong
Setup reporting database	Fred Benbow-Herbert (driver), Julian Day
Barra Interface	Julian Day

Implement core satelite

Plan core satelite take on	Michelle Gray (driver), Jean Luyt, Avril Stassen, Leslie Harvey, Dave Mullord
Develop test plan & environment	Michelle Gray (driver), Avril Stassen
Set up data integrity centre	Dave Mullord (driver)
Ability to control units	Dave Mullord (driver)
Ability to handle complex transactions	Dave Mullord (driver)
Finalise responsibilities	Anton Raath (driver)
User test	Michelle Gray (driver), Avril Stassen, Jean Luyt

Project Planning

Mike and Keith will run the project jointly. It was agreed that each project driver would submit an outline project plan to Keith Rentzke so that he and Mike Kane can synthesise an overall project plan which takes timelines, dependencies and other resources into account. This should be done by the end of the week if possible, but no later that Tuesday.

SUB-PROJECT NAME:			
TEAM LEADER:			
***OVERALL PRIORITY:**			
PURPOSE & SCOPE:			
CRITICAL DEPENDENCIES: (What in your view could cause this subproject to fail?)			
TEAM MEMBERS:			
STEP	**PERSON RESP**	**TARGET DATES**	**#PRIORITY**

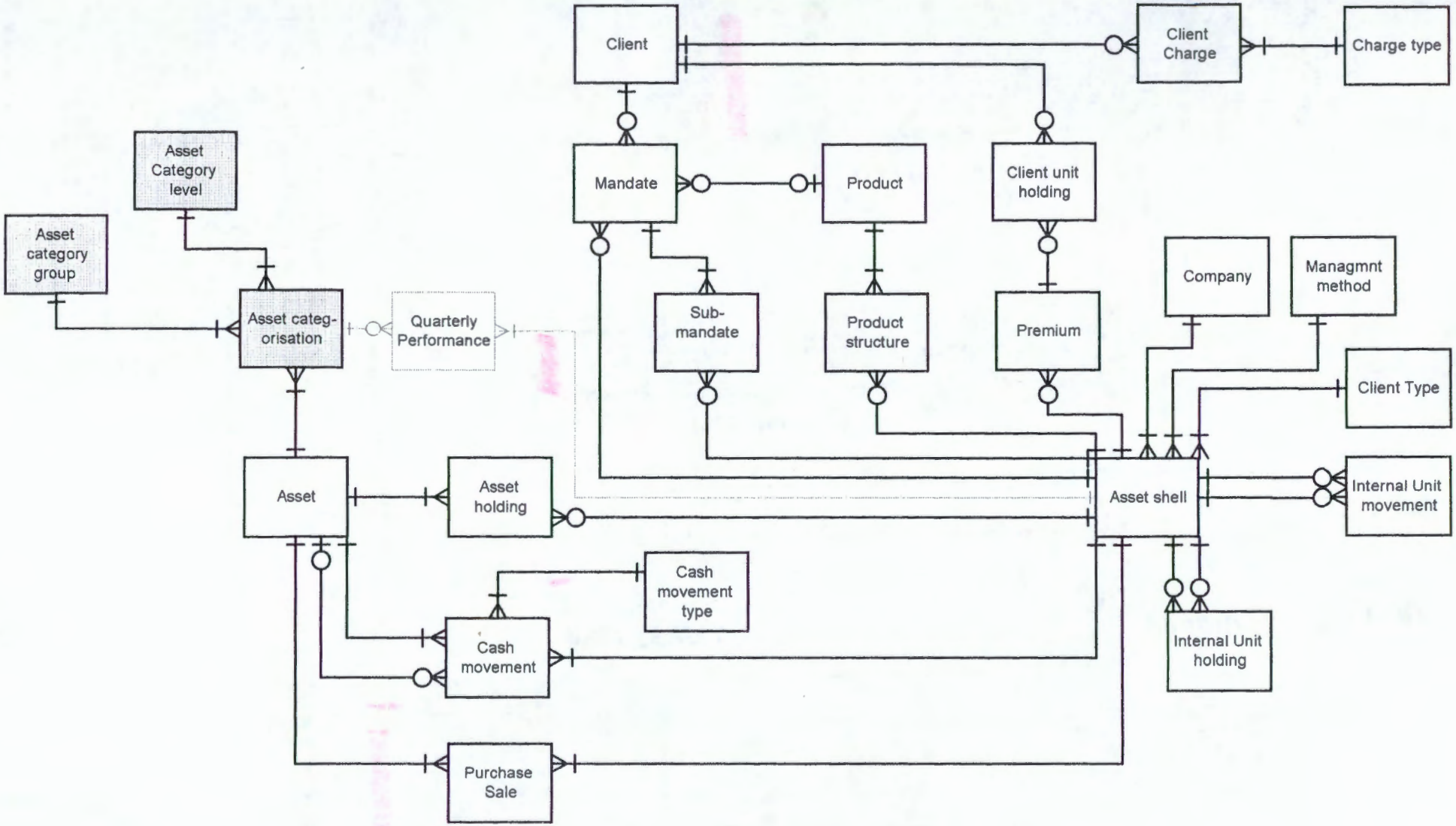
Note:

* = To prioritise your sub-project use the following classification: 1 = Year end requirement, 2 = new investment process requirement, 3 = desirable for new investment process but can be implemented immediately after 1/5/97, 4 = medium term operational, 5 = strategic.

= If individual steps within your sub-project have a different priority to the overall sub-project priority please indicate in this column

B7

APPENDIX C


Client
Client Charge
Charge type
Asset Category level
Asset category group
Mandate
Product
Client unit holding
Company
Managmnt method
Asset categ-orisation
Quarterly Performance
Sub-mandate
Product structure
Premium
Client Type
Asset
Asset holding
Asset shell
Internal Unit movement
Cash movement type
Cash movement
Internal Unit holding
Purchase Sale

C1

C2

CLIENT

ClientID	Client
FR	Fred
JO	Joe

PRODUCT MANDATE

ClientID	ProductID	FromDate	AssetShellId	ProductMandate%
FR	P1	01 January 1997	PF3	50
FR	P2	01 January 1997	PF3	50
JO	P1	01 January 1997	PF3	50
JO	P2	01 January 1997	PF3	50
FR	P1	01 April 1997	PF4	50
FR	P2	01 April 1997	PF4	50
JO	P1	01 April 1997	PF4	50
JO	P2	01 April 1997	PF4	50

CLIENT UNIT HOLDING

ClientID	AssetShellId	EffectiveDate	ClientPremium	ClientUnitHoldingD	TotalClientUnitsToDateD
FR	PF3	01 January 1997	R 1,250.00	1250	1250
JO	PF3	01 January 1997	R 1,250.00	1250	1250
FR	PF3	01 February 1997	R 1,500.00	1500	2750
JO	PF3	01 February 1997	R 1,500.00	1500	2750
FR	PF3	01 March 1997	R 1,500.00	1162.79	3912.79
JO	PF3	01 March 1997	R 1,500.00	1162.79	3912.79
FR	PF3	01 April 1997	R 1,500.00	1162.79	5075.58
FR	PF4	01 April 1997	R 1,500.00	1500	1500
JO	PF3	01 April 1997	R 1,500.00	1162.79	5075.58
JO	PF4	01 April 1997	R 1,500.00	1500	1500

PRODUCT

ProductID	Product
P1	Product1
P2	Product2

ASSET

AssetCode	Asset
EQ1	Equity1
EQ2	Equity2
LC3	LargeCap3
LC4	LargeCap4

ASSETSHELL

AssetShellId	FundName
CR1	Core1
PF3	Portfolio3
PF4	Portfolio4
ST2	Satelite2

PRODUCT STRUCTURE

ProductID	AssetShellId	FromDate	ProductStructure%
P1	CR1	01 January 1997	50
P1	ST2	01 January 1997	50
P2	CR1	01 January 1997	75
P2	ST2	01 January 1997	25

PREMIUM

AssetShellId	EffectiveDate	NetUnitMovement	UnitPrice	PremiumD	TotalUnitsToDate
PF3	01 January 1997	2500	R 1.00	R 2,500.00	2500
PF3	01 February 1997	3000	R 1.00	R 3,000.00	5500
PF3	01 March 1997	2325.58	R 1.29	R 3,000.00	7825.58
PF3	01 April 1997	2325.58	R 1.29	R 3,000.00	10151.16
PF4	01 April 1997	3000	R 1.00	R 3,000.00	3000

ASSET HOLDING

AssetShellId	AssetCode	EffectiveDate	NominalHolding	MarketPrice	MarketValueD
CR1	EQ1	01 January 1997	104.166	R 10.00	R 1,041.66
CR1	EQ2	01 January 1997	104.166	R 5.00	R 520.83
ST2	LC3	01 January 1997	83.33	R 7.50	R 625.00
ST2	LC4	01 January 1997	31.25	R 10.00	R 312.50
CR1	EQ1	01 February 1997	229.166	R 10.00	R 2,291.66
CR1	EQ2	01 February 1997	229.166	R 5.00	R 1,145.83
ST2	LC3	01 February 1997	183.33	R 7.50	R 1,375.00
ST2	LC4	01 February 1997	68.75	R 10.00	R 687.50
CR1	EQ1	15 February 1997	229.166	R 10.00	R 2,291.66
CR1	EQ2	15 February 1997	229.166	R 10.00	R 2,291.66
ST2	LC3	15 February 1997	183.33	R 10.00	R 1,833.30
ST2	LC4	15 February 1997	68.75	R 10.00	R 687.50
CR1	EQ1	01 March 1997	354.166	R 10.00	R 3,541.66
CR1	EQ2	01 March 1997	291.666	R 10.00	R 2,916.66
ST2	LC3	01 March 1997	258.33	R 10.00	R 2,583.33
ST2	LC4	01 March 1997	106.25	R 10.00	R 1,062.50
CR1	EQ1	01 April 1997	604.166	R 10.00	R 6,041.66
CR1	EQ2	01 April 1997	416.666	R 10.00	R 4,166.66
ST2	LC3	01 April 1997	408.333	R 10.00	R 4,083.33
ST2	LC4	01 April 1997	181.25	R 10.00	R 1,812.50

UNIT HOLDING

EffectiveDate	AssetShellId	HoldsInAssetShellID	UnitMovement	MarketValueD	TotalUnitHoldingD	TotalMarketValueD	UnitPrice
01 January 1997	PF3	CR1	1562.5	R 1,562.50	1562.5	R 1,562.50	1
01 January 1997	PF3	ST2	937.5	R 937.50	937.5	R 937.50	1
01 February 1997	PF3	CR1	1875	R 1,875.00	3437.5	R 3,437.50	1
01 February 1997	PF3	ST2	1125	R 1,125.00	2062.5	R 2,062.50	1
01 March 1997	PF3	CR1	1409.77	R 1,875.00	4847.27	R 5,312.50	1.33
01 March 1997	PF3	ST2	922.13	R 1,125.00	2984.63	R 3,187.50	1.22
01 April 1997	PF3	CR1	1409.77	R 1,875.00	6257.04	R 7,187.50	1.33
01 April 1997	PF3	ST2	922.13	R 1,125.00	3906.76	R 4,312.50	1.22
01 April 1997	PF4	CR1	1409.77	R 1,875.00	1409.77	R 1,875.00	1.33
01 April 1997	PF4	ST2	922.13	R 1,125.00	922.13	R 1,125.00	1.22

APPENDIX D

TRANSFORMATION OBJECTIVE AND PRINCIPLES

Asset management is one of the most competitive businesses in the world. The competition ranges around many factors with performance, level of and quality of service and cost of operation being the main three.

With increased international competition in the South African market it is vital that we change in order to be competitive. Due to our history of poor performance and service levels which were (and still are in some areas) lacking, we have more work to do than most of our competitors.

The "Core and Satellite process" of investment management hopefully will address the investment performance problems of the past. In business services we need to change the way we do business immediately and radically to complete the equation. We need to develop into a learning organisation that is comfortable with change and continuous improvement.

To focus the entire division on the change we have decided to set one major objective. There are a number of ancillary objectives which go hand in glove with it.

OBJECTIVE

"Accurate daily unit pricing for all portfolios"

1. This means that all data needs to be checked and, if necessary, corrected daily.
2. We must achieve this objective as soon as possible.
3. The division must be as efficient as possible and benchmarked against local and international competitors for levels of service and costs.

The principles which are set out later deal are guidelines within which we need to design the transformation process. The process will be designed by the people involved in doing the work on a team based approach.

PRINCIPLES

1. Change will be significant. There will be a transformation to re-energise SAM admin, not just a minor shift in direction.
2. Management will be decisive and firm in setting the direction but democratic in working out the practical implementation.
3. The major emphasis will be in designing processes and structures which will enable SAM admin to produce *accurate daily* unit prices at an internationally *competitive* rate.
4. Because an accurate daily unit price is the primary goal, it means that a secondary, but critical goal is to ensure that all input and output data is accurate and checked daily. The unit price process will not be the only process which is focused on.
5. Designing good process and structures will be critical to SAM's success in effecting the transformation. The process design and the structure should mutually support each other, and facilitate good service to the client. We should present "one face" to the client.
6. The first design priority will be to ensure that there is "up front" data integrity, especially for complex transactions.
7. Streamlined processes must be designed so that people are not overburdened.
8. Accountability and ownership will be designed into the processes ("close the loop").
9. Technology can be an effective solution to many problems, but each case will be judged on its merit.
10. Systems supporting the processes must have high integrity (checks and balances).
11. A new work culture must insist on these principles:
    - "Do things right first time - now!"
    - Priorities: Accuracy - first, Speed - second
    - Eliminate duplication and all non value added work
    - We must be open and receptive to new ways of doing things
    - There must be no turf protection
    - We must not harp on history - or waste effort by defending the past

- Improve communication with the 4th floor. We must work closely with them (not against them)
- Performance management systems must support the culture that we want
- Apply the 80/20 rule - put effort where it is needed most and will have most benefit

12. There will be a restructuring into new multiskilled teams, probably based on portfolio types.
13. There will be dedication to increasing skills and business knowledge by:
    - Formal training
    - On the job training
    - Post-mortem analyses
    - Proactive communication
    - Providing a good context for people to work in - a clearer picture of the overall work process and where the individual fits in
    - Structuring meetings so that learning can take place
14. There will be a conscious effort to get the right people into the right jobs.
15. No more staff.
16. There will be dedication to improving communication so that it is effective, appropriately targeted and disciplined.
17. In order to implement these principles, it is important that there must be buy-in from all managers, and also that there is an active, hands-on management style. Action is needed, things must get done, there must be progress.

BUSINESS SERVICES TRANSFORMATION - FOCUS AREAS

BASIC MANAGEMENT

Control, clarity, ownership, accountability, discipline, output
Tighter discipline - Defined procedures - Adherence to procedures
People should take ownership and be accountable for defined outputs
Job descriptions are needed plus proper procedures which need to be followed
Need to get people to take ownership of function - but keep on giving reportback and feedback
Develop a work output culture (fuzziness lets people off the hook, the whole business is too loose)
Consultants need to be given clear deliverables and empowered to carry them out eg:
- map process flows clearly
- restructure admin
- develop and document clear, tight procedures

Separate user functions from IT functions (IE Jean and Michelle's dual admin/IMPART role)

Admin cohesiveness
Dave, Butana, Doreen need to link up into a more cohesive unit
Tight control of dealing sheets, the key input to admin
Portfolio admin needs to be managed - there is no control or teamwork

Stronger admin management
Stronger, more active management needed - a bit more autocratic as people have too much latitude
Butana and Andy should be replaced by a strong manager
Must get right type of managers:
- able and willing to sort out problems
- interact more with staff
- have a better day to day handle on things

Admin needs stronger management - clear direction, less fickle, more consistency
- If Butana goes, it will be a disaster
- Spend less time doing work, more time managing
- Crisis management causes as many crises as it solves
- Everybody should receive fair and equal treatment

Need firm management

Develop a professional attitude to work
There is a general lack of professionalism
- Much telephone talk, banter, radios playing, messing about in data capture when boss is out
- The environment is not productive - people do not concentrate on the job
- No wonder there are data errors

BUSINESS COHESION

Tighten up the operation
Decisive, controlled, effective decision making
Skilled people should do up front work
Tighten structures, definitions of terms and concepts, and the way meetings are run
We need to find a way to counteract the over-emphasis on marketing within SAM

Develop big picture thinking
Tighten up the "middle view" of the business - middle managers layer - there is a missing link.
Consolidate a combined up front view
People only want to get involved with their little bit of the job
- Not worried about the final out come
- Not interested in grabbing problems and trying to sort them out
- Admin must be one big team (first) - smaller teams (second)
- Crossover from one area to another is poor, teams must work together
- Success will reflect in the unit price

People need a better understanding of the financial disclosure aspects of SAM's business
- A bigger picture up front would enable financial disclosure to be more cost effective
- People need to see the implications of their work down the line

History - it is important to remember where we came from - why are things done in a certain way
- Mona can often answer these questions - but many can not be answered
- Mona is able to change because she can see the reason for it - she understands the history
- People who do not know why they are doing something are stuck - no insight
- Question core assumptions

Develop cohesion via effective software process

Develop and instil throughout SAM a workable software process
Need to develop an insight into the software process (plus a more meaningful name for it)
Must not have a mentality that every problem can be solved by throwing technical solution at it:
- This does not just mean computer systems but things like PPM
- Management must take the lead - too many technical solutions flop
- We must use our own common sense

Lead via IT - automate as much as possible - reduce manual work as much as possible

Cohesion via better communication

Communication is poor - no confidence that a complex job will go all the way through accurately
Communication about change is the big problem
- Consultants must keep everyone in the picture not just the big bosses
- People on the floor, including middle managers, need to be kept in the picture

Team spirit was good during core/satellite - better 3rd floor and 4th floor communication
Biggest problem is the 4th floor - bad vibes - but how to stop them?

DATA INTEGRITY

Get things right up front

Set things up right vs painful checking after the event
Pre-validation vs query solving after the event
Improve IMPART data capture (esp complex transactions like asset take on)
Correct IMPART transaction (eg purchase and sales vs book cost adjustments)
Must get integrity on capital events
Critical thing to change if we want to get an accurate unit price is up front data integrity into IMPART

Responsibility for quality

All portfolios should be checked every day - especially accounting transactions
There is a tendency for people to hide mistakes in admin.
IMPART data capture should take responsibility for the consequences of their work
Less data fiddling on IMPART

Tighter control of complex transactions

Complex transactions need a lot more tightening up - still get bad data coming to data capture.
Up front data capture and complex transactions need more work

Clarify the term "data integrity" - where is the problem?

Confusion about what is meant by "poor data integrity" needs to be cleared up. Is it:
- Data capture?
- Impart processing?
- Input to data capture?
- Some combination of the above?
- How come we had such a good year end if data integrity is so bad?

Very rough, thumbsuck distribution of problems:
- Up front - pre data capture - 20%
- Data capture - 50%
- Impart - 20%
- Back end - reporting etc - 10%

UNIT PRICING

Business transformation objective - cost effective quality
Must get accurate daily unit prices
Objective - accurate daily unit price is the right objective
- Especially being cost effective - it is expensive not to get things right first time
- If you are fixing mistakes, you are losing money

Blue files must go as quickly as possible
There must be a focus on unit pricing - training plus cohesiveness plus up front data capture

Ownership of unit pricing
Resolve who will take ownership of unit pricing - who will check and validate and do the work
Unit pricing controls the admin operation and is the area to focus on
- Jean is the only one who knows unit pricing, how it is structured
- Jean can manipulate the unit price
- Can only do certain things in certain ways for certain reasons because of unit pricing
- It is frustrating that all work revolves around unit pricing

IMPART data capture should feel more pain re their contribution to incorrect unit prices

Training in unit pricing
There must be extensive across the board training in unit pricing

INFORMATION TECHNOLOGY

Major IT projects:
unit pricing
performance measurement and attribution
automated dealing
reporting and model office month end
Performance attribution *must* be implemented

Improve systems quality
Systems do not cater well for finance - need improved accuracy, reports, layout, presentation of data
- Some old reports are not available, not in the right format, work arounds are needed
- Better checking before data is handed to finance
- IT need a better understanding about what finance does
- Biggest headache is the systems
- Sometimes takes a long time to get a run out, operation errors, high paper costs

Do more systems validation - IMPART as a engine, IT writes validation input screens, batch controls

Develop an IT strategy
Set up an IT steering group
A view of the future is needed - what will happen to IMPART - a software strategy
Datawarehouse is needed - too much happens in EXCEL
4th floor should be allowed to use tools if they like - but should subscribe to good IT controls,
- good methods, documentation, and standards
- there is a worry that IT will pick up a mess - no continuity or insight - no documentation

Clients must be linked electronically to our systems - they can then do "what ifs" themselves
One way to reduce non value added work is to make people more aware of the tools on their desktops
- People do a lot of manual reworking of data that could easily be automated
- Training in Excel, a few tips, asking more questions - people could work a lot smarter

Worry about falling behind our competitors as far as technology is concerned
- Also worried that we do not have the right resources for say a big system rewrite

It is difficult to plan in the IT area as complex work with tight deadlines is thrown at you
- Often this work is not thought through - just dumped with no warning - at last minute
- Development work suffers
- IT people should do IT work not admin work - like checking etc

Data capture could not be turned into a dumb operation - do a lot of other work - recons, checking, etc

STRUCTURE 8

There should be a rigid pipeline, production line which can cope with routine work
In fact there already is one - Data Capture works well - It should be extended to cope with more
Data Capture is well run, well managed, people know what they are doing - you can work with them
Only problem is complex transactions - the most important thing to get right
- Data capture can not tell if they are being given garbage
- They make the odd mistake
- They process the garbage (from complex transactions) accurately

Complex transactions should be project managed
- pointless trying to prescribe a set procedures - they will never be right
- need a way of linking up 4th floor and plugging things into production line where possible
- see the complex transaction through - think out all the twists and turns up front
- the people who manage complex transactions should be outside the rigid production line

People in production line need to know their role, process and procedure, and see the bigger picture

STRUCTURE 9

No specific opinion except that the big problem is to do with 3rd floor 4th floor dynamics
- Dealing not up to scratch
- There is a club attitude - attack is best form of defence - destroys team spirit

But main thing is 3rd floor is starting to win
Privacy is important - seating arrangement - not having someone breathing over your shoulder

STRUCTURE 10

Structure is fine in my area - and not certain enough to comment about the other areas
Should not rush into radical change - lets see how the new management get on

STRUCTURE 11

Team idea needs exploring - there are lots of possibilities
Important that a link is built to the 4th floor - a way of linking each team with 4th floor
- Link to investment team
- Link to Business development
- Perhaps to Portfolio Structure Group, SMP

Explore possibility of a team doing everything from dealing, data capture, portfolio admin, client liaison
- Then they can take responsibility for things
- See whole process, understand whole business, get a way from compartmentalised view

Get away from lots of little black boxes who can not see beyond their own area
Might not be a good thing to structure on a portfolio basis - inhibits free movement from team to team
Teams may introduce more competitiveness

STRUCTURE 12

Teams of 4 people work well - easy to manage - enough back up
Team leader is at a degree level - say grade 14
Members can back each other up - do everything that the team is responsible for
Process should be clearly mapped, with fairly tight job descriptions - work flow idea - paper based
Metrics so they can see how they are performing - rewarded accordingly
See the job through from start to finish
Can learn a lot from the SUTCO/IMC experience
Brent can help with adjusting to change
Teams should have an equal number of functions to perform even if they are not the same functions
- Equal work load - equal skills needed -equal complexity - if possible

Recons done within the team and good result handed on - eg to finance

Must not let analysts come in between IT programmers and the users
programmers must speak directly with the users
Mainframe approach is not suitable in a PC environment
The projects are not big enough
No good having analysts who do not understand the tools or the business

Improve teamwork and professionalism within IT
Not easy to think about change when you are feeling positive! IT area is now running well.
More communication needed within the IT area - progress meetings
IMPART team is not professional and work quality is not good enough
Loose, slack attitude, not enough quality checking, more discipline needed
Technical area gives mixed service
There is turf protection around IMPART
Jean and Michelle are the only ones with knowledge
This makes the learning environment difficult and inefficient
Work time is lost during the day by lateness and non-work activities
IMPART controls how work gets done
Caught in a spiral of IT people unable to shed admin work so development work suffers

SERVICE

Develop a service culture
Develop a "get it right first time culture"
Culture of wanting to take on problems - not running away from them

Develop world class service
Develop superior client reporting - better info, more accurate, timeous
Develop internal reporting - eg SAM profitability
Develop enhanced reporting for Southern Life
Develop ability to take on new business easily and efficiently
Our service must be cost effective - leaner, better margins within two years so we can compete
Admin should not just be a record keeping operation - more client liaison
Find other ways of making money
SAM is too used to being "thumped" - part of SAM's history of poor performance:
This means we carry on getting thumped
We get worse service than other asset managers -eg settlements from Brokers
Do work for both Portfolio take ons and take offs - hammered from both sides
Still working a year later on lost business - satisfying clients, auditors, new asset manager
Must not be hamstrung by our history -
Must demand good service
Must not do non value added work

PEOPLE MANAGEMENT

Training and career development
Must develop a capacity to train - potential trainers are key producers so have no time to train
Concern that it will not be possible to stick to career plans and promises because of transformation
Portfolio admin were thrown in the deep end, with little training. How can they be good at checking?
Training is important - but difficult to train and produce at the same time
Increasing complexity exacerbates this - many jobs are too complex to be done by clerks
Training is important - courses - talks - people want to know the business
Feel sympathy for people thrown in the deep end - lacking knowledge - making mistakes - afraid to ask
Investments is a complex business - learning is on-going - better training could help
Especially giving data capture and systems people insight into the financial area
Career planning and management is poor - no direction
The admin area needed better training around IMPART

Resistance to change

Break down attitude of "my turf" (eg "my people" in data capture)
Need to manage resistance to change - people are stuck - we do not know how to manage change
It is not the new things which cause the problems - things which go straight on to IMPART work fine
- It is changing the old stuff into the new where things go wrong
- Management of change - hanging on to the past is an issue
- People do not believe the blue files will go
- Must not hold back on change for the last 1% which is not right - let the past go - move on

Appropriate skills for the job

We must not blame the people - there are few bad apples - only bad management
Portfolio management does not have the right calibre of people - skills are missing
Get the right people in the right job
People management should be directed toward growing natural teams:
- Get good people and know what turns them on
- Adapt the job to fit them so that they can be enthusiastic about it
- Provide right training so that they are able to satisfy the customer
- Allow natural teams to grow and mesh (vs management declaring people to be a team)

Important to get cost down - best way is to get rid of dead wood - suppose by natural attrition over time

Values

Need to look at peoples values ("build a better life by stealing office supplies")
It is not right that people should have to work so much overtime
Constructive, adult to adult performance appraisals should be re-introduced
Be open about things like bonuses, arrangements with people etc
- Why hide things - people talk anyway- it comes back to haunt you

12 OPINIONS ON A POSSIBLE NEW STRUCTURE

STRUCTURE - 1
Unite Portfolio Admin and Impart (IE unite Butana and Doreen's functions)
There is no linkage between the two areas and there should be
A team will be responsible for a certain type of portfolio
Deal sheets will be split accordingly
Team will be responsible for quality service and data re:
- Unit price
- Ledger
- Portfolio managers
- Client reporting

STRUCTURE - 2
Enforce a process based approach
Multiskilled teams based on certain types of portfolio
Dealing automated
Do data capture, portfolio admin, valuation, unit pricing, reports (incl finance reporting)
Managed by specialised, knowledgeable client liaison officer (graduate level)
There may be specialised central functions like settlements, capital events
Finance and IT would be separate

STRUCTURE - 3
Customer focused teams - multiskilled
Based on portfolio type to lessen amount of skills needed in each team
Portfolio admin and data capture all in same teams

STRUCTURE - 4
Teams based on portfolio type for client reporting and liaison, portfolios recons and dogsbody work
Supported by automated dealing, Accounting module providing a ledger link, datawarehouse
Following central functions: income control, unit price control, deal checking
Middle office set up for complex transactions and 4th floor link

STRUCTURE - 5
Self-contained teams based on portfolio type
As much automated as possible - eg turn capture clerks into validation clerks - automated dealing
Daily verification - esp unit price
Team does everything including client liaison, tax, and legal
Central control for capital events and custodianship - Joburg office centralised in Cape Town
Training done by Jean, Leslie, Butana, Rob

STRUCTURE - 6
Long term - process based approach - basically current operation glued together more tightly
Restructuring will not change nature of work - functions will still stay the same
Key is to do the current functions more efficiently
More admin staff needed
Marketing and admin need to speak with one voice - tighter bond
More contact with clients

STRUCTURE 7
Not averse to teams in principle
Data Capture remains separate. Too scary to have everyone able to capture data on IMPART
Teams will do portfolio administration - main purpose to check everything on a daily basis
Check - accounting transactions, unit price, make sure everything is in sync and balances
Some of the ideas are not practical - eg data capture in multiskilled teams:
- There would have to be massive training on IMPART
- On what basis and by what process would things be allocated, split amongst portfolio teams?
- Redemptions, partial settlements etc would run across teams - need some up front sorting out

Data capture could not be turned into a dumb operation - do a lot of other work - recons, checking, etc

STRUCTURE 8

There should be a rigid pipeline, production line which can cope with routine work
In fact there already is one - Data Capture works well - It should be extended to cope with more
Data Capture is well run, well managed, people know what they are doing - you can work with them
Only problem is complex transactions - the most important thing to get right
- Data capture can not tell if they are being given garbage
- They make the odd mistake
- They process the garbage (from complex transactions) accurately

Complex transactions should be project managed
- pointless trying to prescribe a set procedures - they will never be right
- need a way of linking up 4th floor and plugging things into production line where possible
- see the complex transaction through - think out all the twists and turns up front
- the people who manage complex transactions should be outside the rigid production line

People in production line need to know their role, process and procedure, and see the bigger picture

STRUCTURE 9

No specific opinion except that the big problem is to do with 3rd floor 4th floor dynamics
- Dealing not up to scratch
- There is a club attitude - attack is best form of defence - destroys team spirit

But main thing is 3rd floor is starting to win
Privacy is important - seating arrangement - not having someone breathing over your shoulder

STRUCTURE 10

Structure is fine in my area - and not certain enough to comment about the other areas
Should not rush into radical change - lets see how the new management get on

STRUCTURE 11

Team idea needs exploring - there are lots of possibilities
Important that a link is built to the 4th floor - a way of linking each team with 4th floor
- Link to investment team
- Link to Business development
- Perhaps to Portfolio Structure Group, SMP

Explore possibility of a team doing everything from dealing, data capture, portfolio admin, client liaison
- Then they can take responsibility for things
- See whole process, understand whole business, get a way from compartmentalised view

Get away from lots of little black boxes who can not see beyond their own area
Might not be a good thing to structure on a portfolio basis - inhibits free movement from team to team
Teams may introduce more competitiveness

STRUCTURE 12

Teams of 4 people work well - easy to manage - enough back up
Team leader is at a degree level - say grade 14
Members can back each other up - do everything that the team is responsible for
Process should be clearly mapped, with fairly tight job descriptions - work flow idea - paper based
Metrics so they can see how they are performing - rewarded accordingly
See the job through from start to finish
Can learn a lot from the SUTCO/IMC experience
Brent can help with adjusting to change
Teams should have an equal number of functions to perform even if they are not the same functions
- Equal work load - equal skills needed -equal complexity - if possible

Recons done within the team and good result handed on - eg to finance

TRANSFORM BUSINESS SERVICES

AGREED PRINCIPLES

Criteria and guidelines

DESIGN:

Process

DATA INTEGRITY
Get things right up front
Responsibility for quality
Tighten control of complex transactions
Clarify the term "data integrity" - where is the problem?

UNIT PRICING
Business transformation objective (cost effective quality)
Ownership of unit pricing
Training in unit pricing

Structure

STRUCTURE

12 structure opinion for ideas and consideration

Mutually supporting

Necessary enablers

Necessary outcomes

OBJECTIVE: "Accurate daily unit pricing for all portfolios"

BASIC MANAGEMENT
Control, clarity, ownership, accountability, discipline, output
Admin cohesiveness
Stronger admin management
Develop a professional attitude to work

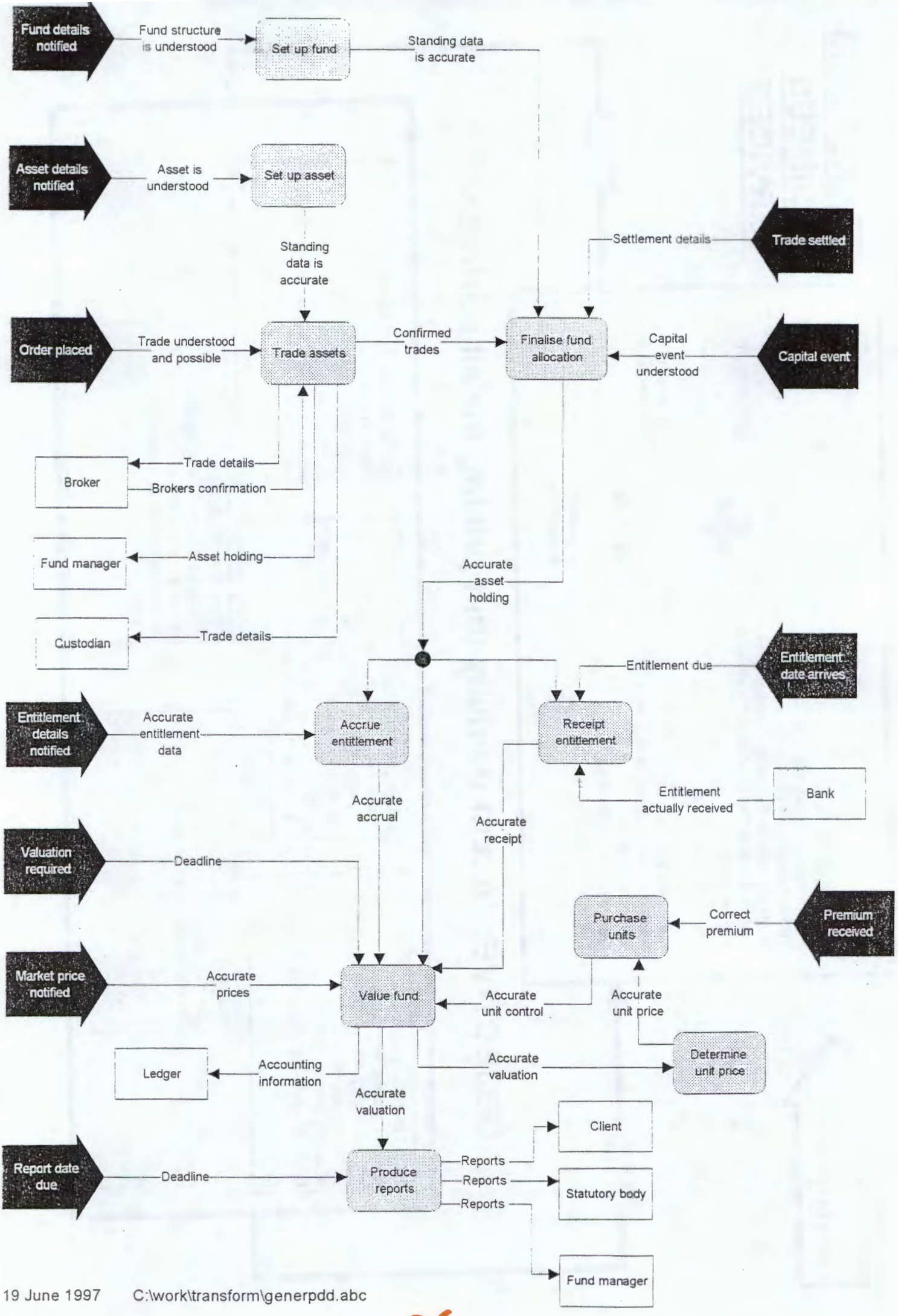
BUSINESS COHESION
Tighten up the operation
Develop big picture thinking
Develop cohesion via effective software process
Develop cohesion via better communication

INFORMATION TECHNOLOGY
Major IT projects
Improve systems quality
Develop an IT strategy
Improve teamwork and professionalism within IT

SERVICE
Develop a service culture
Develop world class service

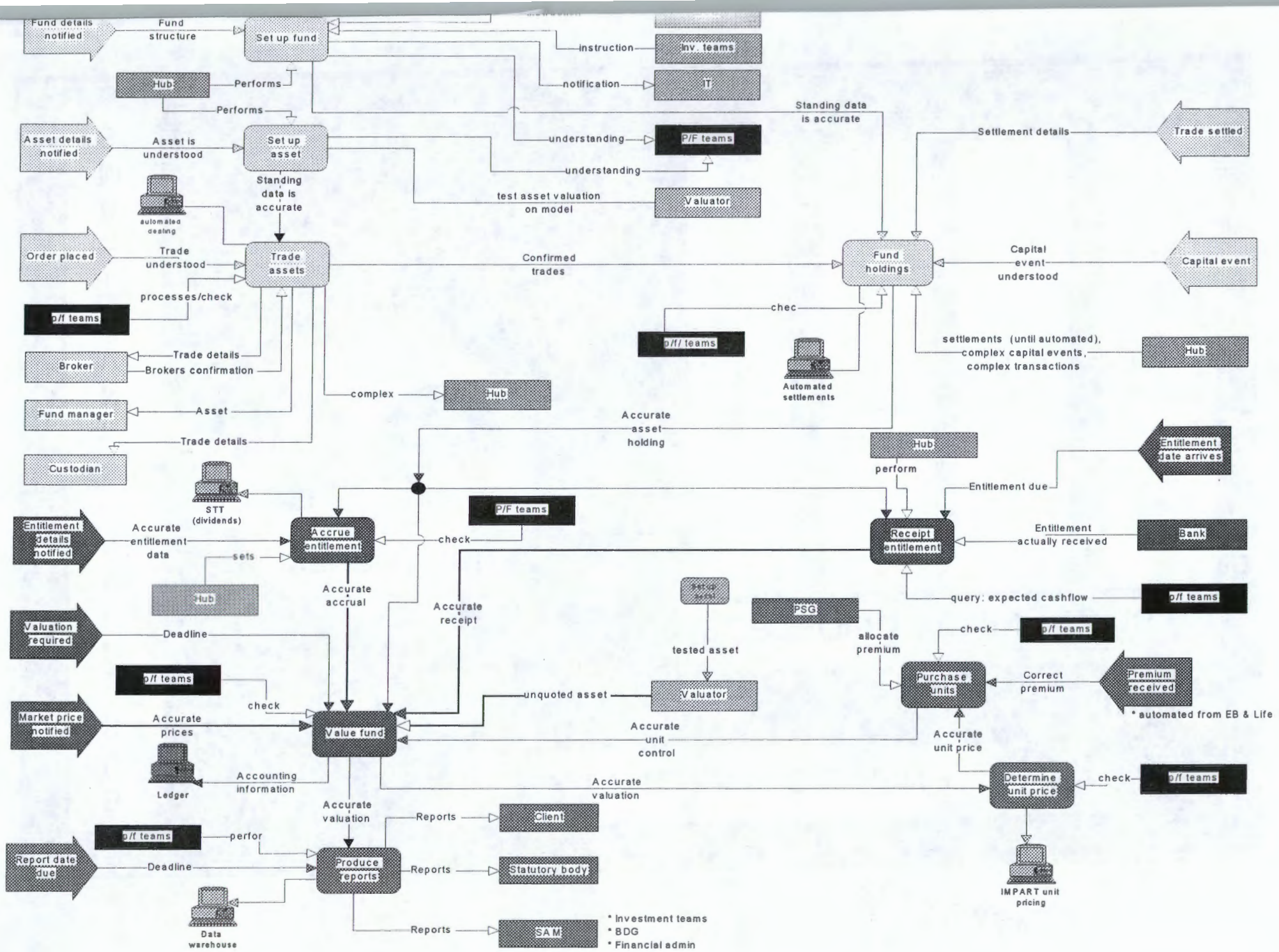
PEOPLE MANAGEMENT
Training and career development
Resistance to change
Appropriate skills for the job
Values

04



19 June 1997 C:\work\transform\generpdd.abc

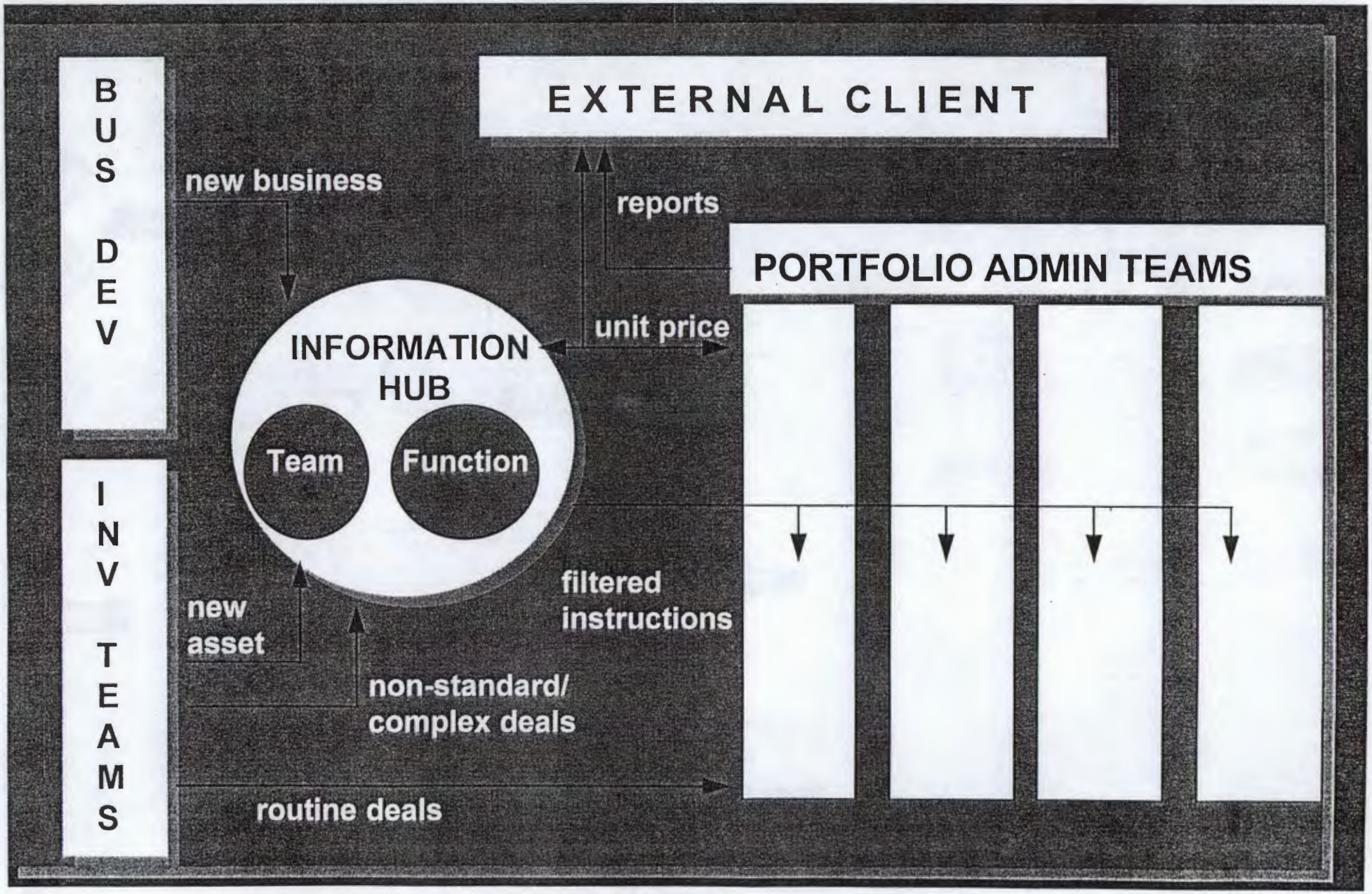
DS


Fund details notified
Fund structure
Set up fund
instruction
Inv. teams
Hub
Performs
notification
IT
Performs
Standing data is accurate
Asset details notified
Asset is understood
Set up asset
understanding
P/F teams
understanding
Standing data is accurate
test asset valuation on model
Valuator
automated dealing
Order placed
Trade understood
Trade assets
Confirmed trades
Fund holdings
Capital event understood
Capital event
processes/check
p/f teams
chec
p/f teams
Automated settlements
settlements (until automated), complex capital events, complex transactions
Hub
Broker
Trade details
Brokers confirmation
Fund manager
Asset
complex
Hub
Trade details
Custodian
Accurate asset holding
Hub
perform
Entitlement date arrives
Entitlement due
STT (dividends)
P/F teams
Entitlement details notified
Accurate entitlement data
Accrue entitlement
check
Receipt entitlement
Entitlement actually received
Bank
sets
Hub
Accurate accrual
Accurate receipt
query: expected cashflow
p/f teams
PSG
Valuation required
Deadline
tested asset
allocate premium
check
p/f teams
p/f teams
Purchase units
Correct premium
Premium received
* automated from EB & Life
Market price notified
Accurate prices
check
Value fund
unquoted asset
Valuator
Accurate unit control
Accurate unit price
Ledger
Accounting information
Accurate valuation
Determine unit price
check
p/f teams
Accurate valuation
Reports
Client
p/f teams
perfor
Report date due
Deadline
Produce reports
Reports
Statutory body
Data warehouse
Reports
SAM
* Investment teams
* BDG
* Financial admin
IMPART unit pricing

90

SAM ADMIN TRANSFORMATION PROJECT

Kick off Meeting



LO

HUB

- Owns the data integrity of assets
- Acts as a "knowledge filter" for all non-standard transactions
- Understands & instructs teams on complex transactions and the impact on portfolio's
- May consist of teams
- Will house all centralised functions
- Supports and provides direction to the portfolio admin teams

PORTFOLIO ADMIN TEAMS

- Own the data integrity of a specific group of portfolios, which have been selected on similar structure and client requirements
- Receive direction from the hub in respect of non-standard transactions
- Understand how each transaction impacts on the portfolios under their administration
- Consist of multi-disciplinary teams
- Are responsible for checking all transactions & data in the portfolio
- Is responsible for delivery to the client in respect of reports and information

APPENDIX E


LINDSAY R


MELANIE


WENDY


SHALEEN


JAMIE


DOREEN


JAMES G


KOBUS


ARNO L


ROY


MICHAEL J


IMPART


DARYL C


Fred


GUY


REZA


P/F ADMIN


DST

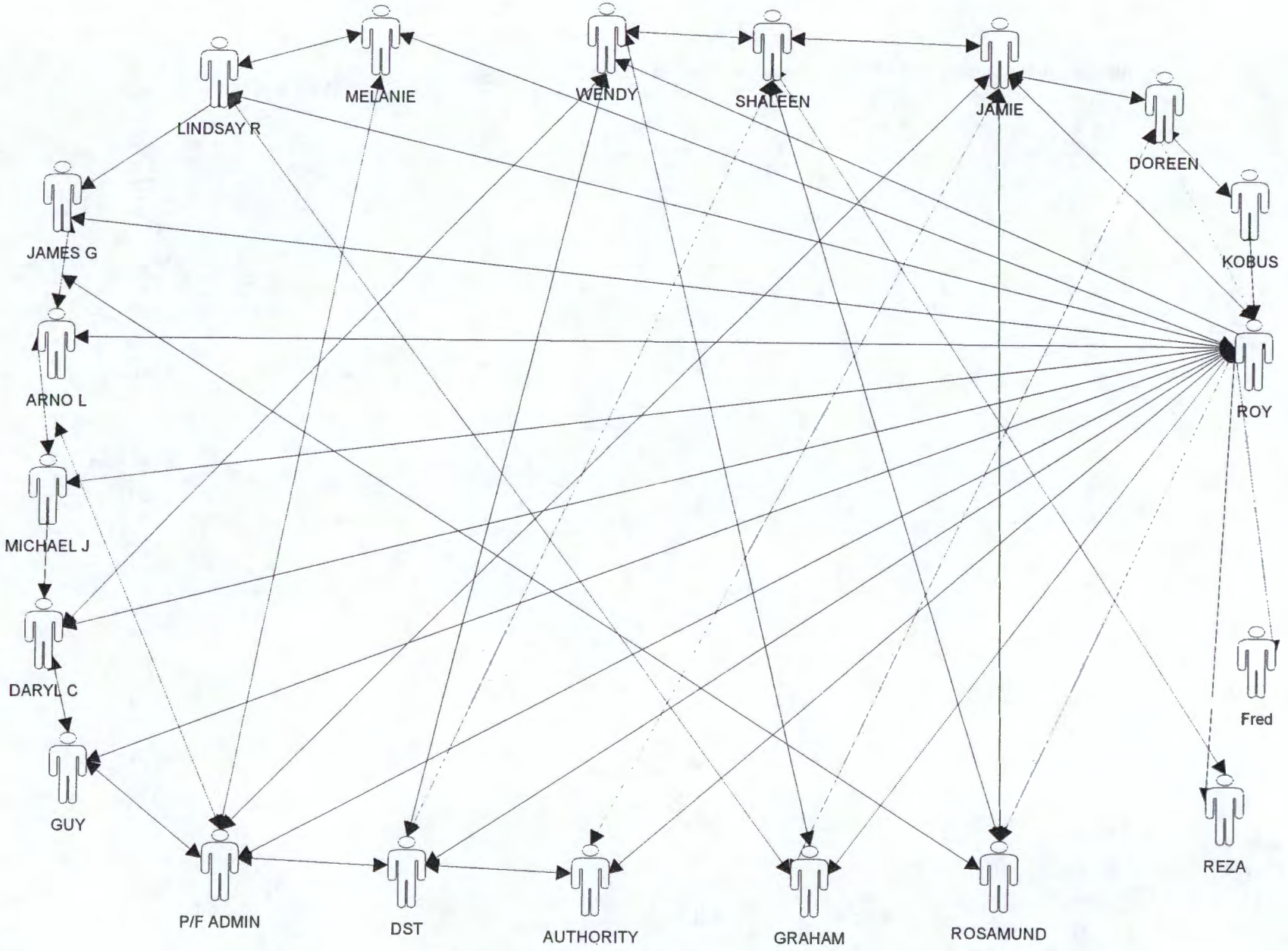

AUTHORITY

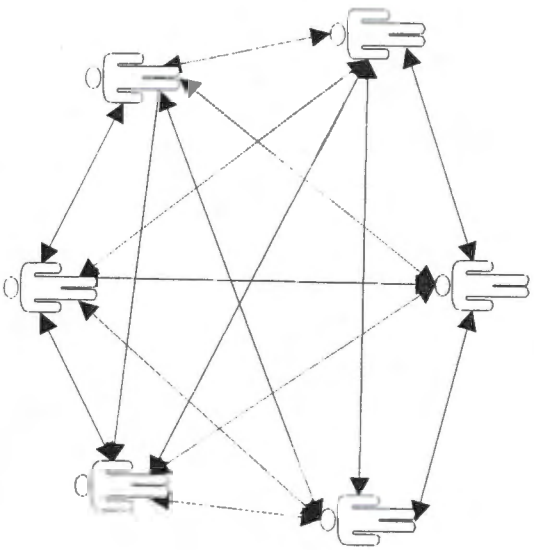

GRAHAM

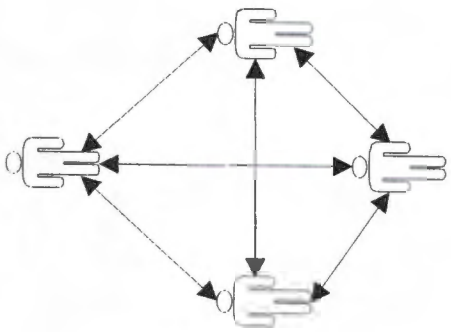

ROSAMUND

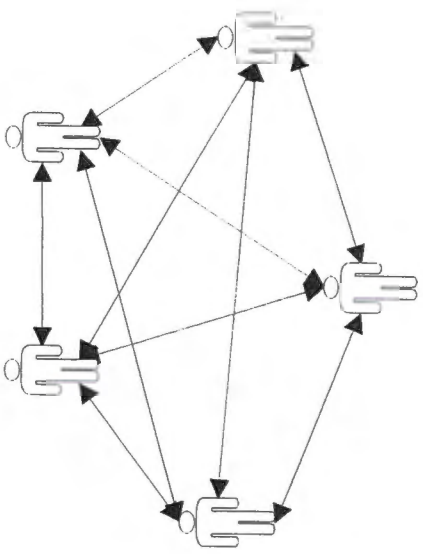
E1

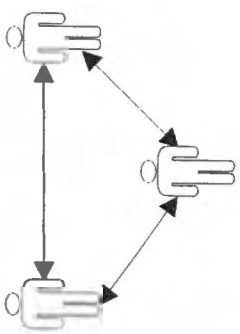
E2


LINDSAY R
MELANIE
WENDY
SHALEEN
JAMIE
DOREEN
KOBUS
JAMES G
ARNO L
ROY
MICHAEL J
DARYL C
Fred
GUY
REZA
P/F ADMIN
DST
AUTHORITY
GRAHAM
ROSAMUND



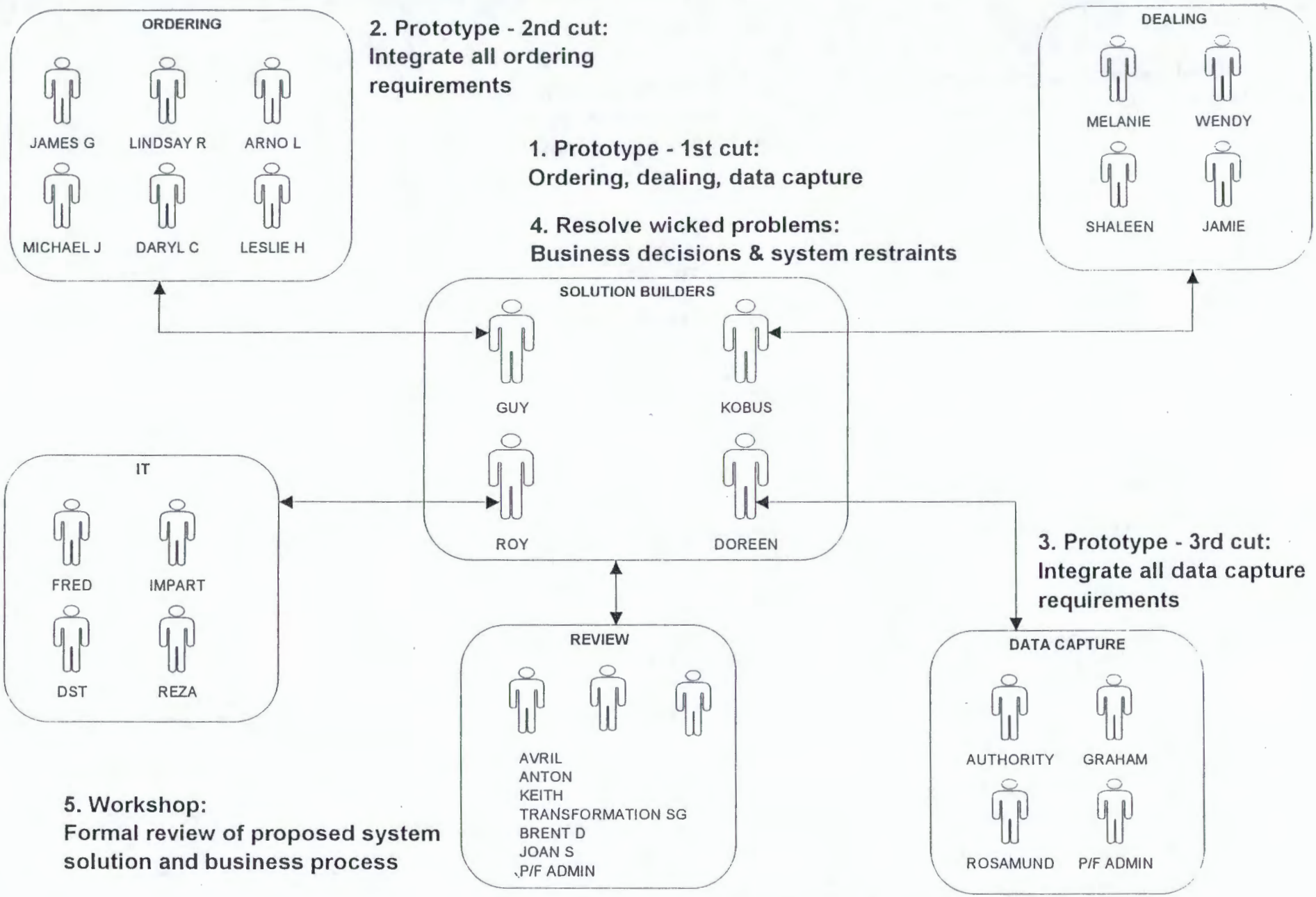




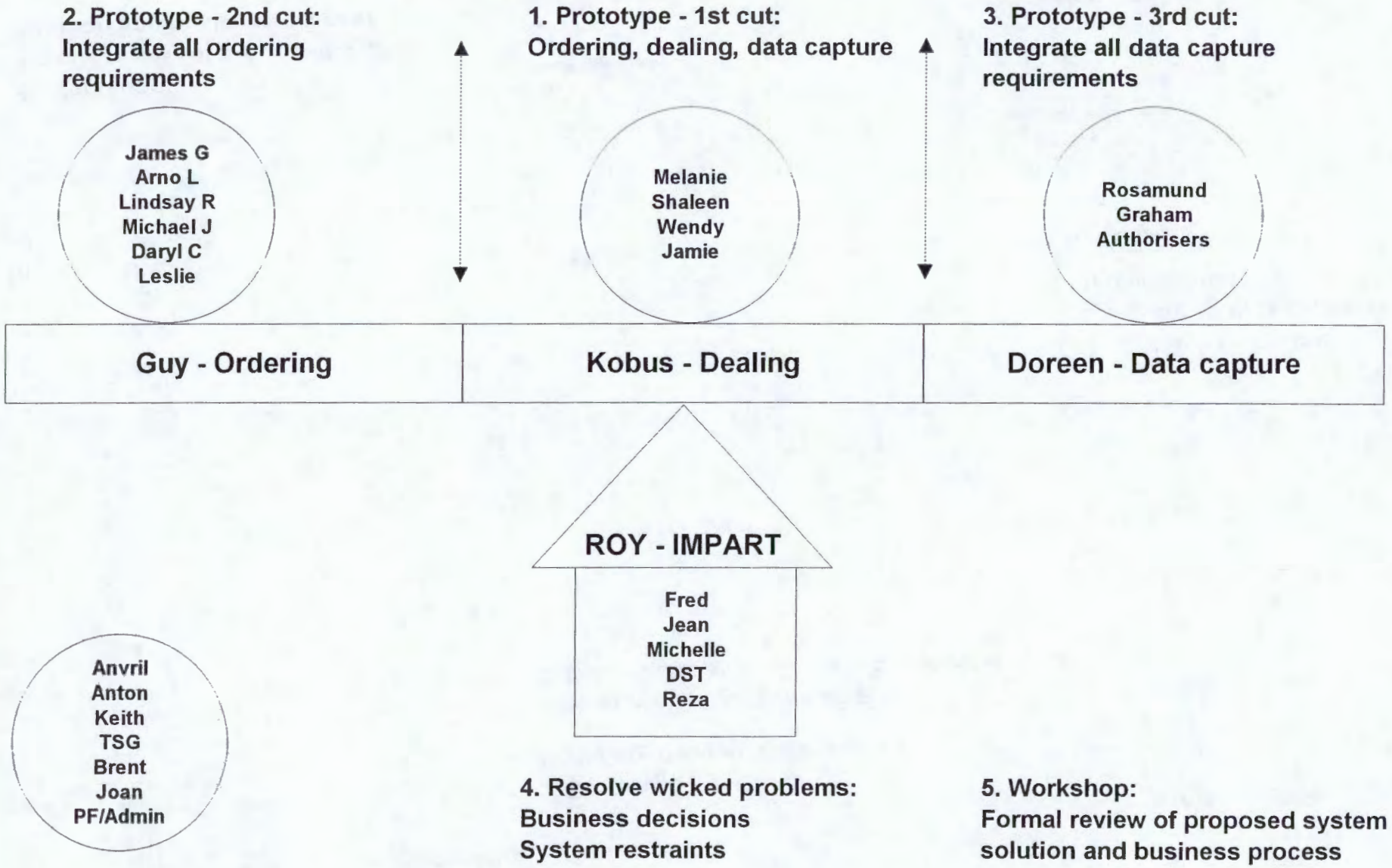


E3

E4


ORDERING
JAMES G
LINDSAY R
ARNO L
MICHAEL J
DARYL C
LESLIE H
2. Prototype - 2nd cut:
Integrate all ordering
requirements
DEALING
MELANIE
WENDY
SHALEEN
JAMIE
1. Prototype - 1st cut:
Ordering, dealing, data capture
4. Resolve wicked problems:
Business decisions & system restraints
SOLUTION BUILDERS
GUY
KOBUS
ROY
DOREEN
IT
FRED
IMPART
DST
REZA
REVIEW
AVRIL
ANTON
KEITH
TRANSFORMATION SG
BRENT D
JOAN S
P/F ADMIN
3. Prototype - 3rd cut:
Integrate all data capture
requirements
DATA CAPTURE
AUTHORITY
GRAHAM
ROSAMUND
P/F ADMIN
5. Workshop:
Formal review of proposed system
solution and business process

DEALING PROJECT - MODEL OFFICE PHASE



APPENDIX F

PROPHET PROJECT INTERVIEWS - 30/6/98 to 2/7/98

Project management and roles
Roles are unclear. "Muddied roles".
Lack of co-ordination.
I was the project manager, but when I came back from holiday it was taken away.
Its my project. It was given to me to do.
No management or communication of expectations.
Lack of focus on Prophet. Other crises seem more important. People are pulled off the project.
No measures of performance.
Direction is needed.
We need one and *only* one project manager.
We need a more defined goal. Goals are too far away. There should be more immediate deadlines.
There should be a combination of a carrot and a stick to help motivate us.
There must be fixed commitments to deadlines.
There is no real team leader. Everyone is on their own mission. There is a lack of overall cohesion.
The project just drifts along.
Directives are vague.
Project management relies on Jeremy, but progress management is done by John. This can not work.
Since last valuation, March 1998, deadlines are fuzzy.
There is unclear accountability.
Management should trust and let go. Give responsibility.

Demotivation, low management trust and support
Our positive ideas are squashed.
There is little management support for this project.
John is a demotivating influence. His most used phrase is "I have some concerns about ....."
John is a worrier and squashes ideas.
I am demotivated.
My motivation goes up and down.
They did not consult us about new appointments to the team.
They should ask not tell. For example, how do they know what skills are needed in the team?
They have a "dash our hopes" approach.
They ask for responsibility but do not give any responsibility.
The takeover is unsettling. Are we going to be retrenched?
It took three or four years for this department to get serious about Prophet.
I am working on old products. These are not seen as a high priority.
Prophet got off to a negative start. This has continued throughout the project.
The team needs more support.

Purpose
Nobody talks and acts as if the project is important.
The project seems low key. It does not seem important.
What is the consequence of not getting the deadline?
If the project is important, why doesn't management protect it from other interruptions.
I can't see the big picture. What is the context of the project? How big is it? How many products?
How does our project relate to the Prophet project in the company that is taking us over?
Why commit to deadlines? There is no real penalty for missing them!
We can work out what the 'real' deadlines are.
Deadlines are not real. We always work to the actual deadline, not the one we are 'given'.
Crunch time is ahead, yet they drift along unconcerned.

Positive Motivation
I like Prophet - it is a good system and internationally recognised.
We can enjoy working together.
Learning about Prophet is exciting.
I like the learning environment, the questioning and digging skills needed for this project.
I get excited about risk - I want to get the job done - this is the priority - to do it.
Prophet is interesting. But after a while, I can imagine that it could become boring.

Prophet skills are internationally recognised - it is good to have Prophet skills on your C.V.
I felt good when I implemented my first product on Prophet. I told my wife about it when I got home.
I feel sure we can do the whole project well before the deadline.
I am enjoying the project. It will look good on my C.V.
This team is capable of self-discipline.
I want to get more involved with Prophet.
We can be better than Takeover Company.

Teamwork
We should talk things out in the open.
We should be involved in team appointments.
We should have informal reviews (accompanied by tea, coffee and cake).
Team sprit is down because we don't have any team functions or events.
We worked two weeks off-site to at Mark's house. We got a lot of work done and it was great fun.
Maybe there are too many people on the project? Do we have too many or too few people involved?
Team culture - we need to be excited.
There is a clash between our team style and our team structure.
The team itself decides on how to assign the work.
The friction is caused by insecurity.
Some people, like Bjorn, need structure while others like MAP are secure in themselves.
There is a general lack of trust. There is no sense of community - not with John as manager.
The team should talk about its insecurities and face up to them.
I got the business logic manual going. It is in living use and supported by the whole team.
I respect Jeremy.
We should have more off-site "power weeks" where we are all focused on Prophet as a team.

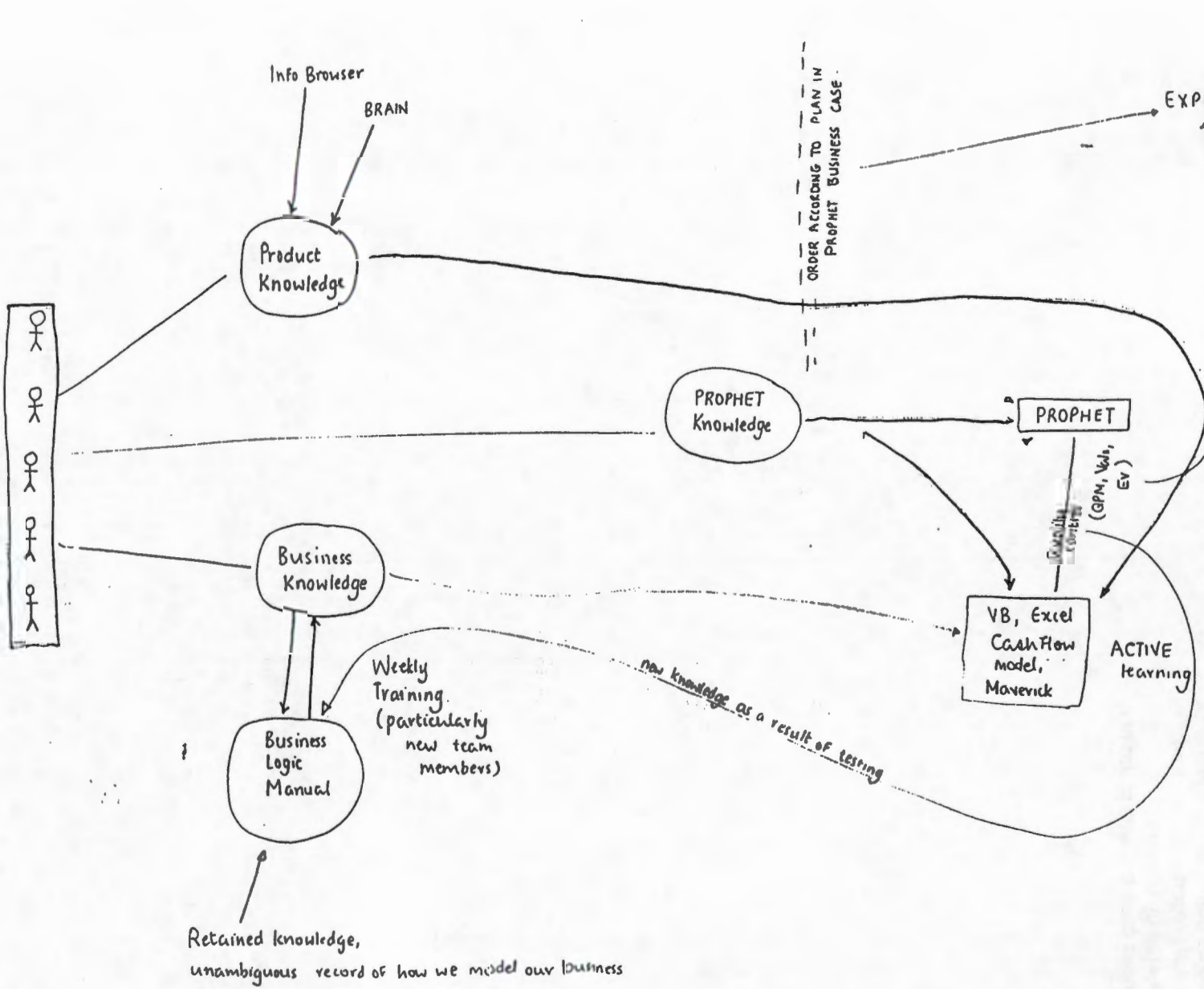
Quality, learning, documentation
We have always emphasised team learning and quality.
I got the business logic manual going. It is in living use and supported by the whole team.
The business logic manual enables the spread of product knowledge.
The business logic manual is a great help.
My role is to provide data and test for quality. When people say "I need data", I give it to them.
We are sticklers for quality - everything balances to the last cent.
We test for identical results between Prophet and Maverick.
There is a well recognised testing strategy.
My perception is that Takeover Company cut corners.
Testing is rigid.
Even Internal Audit agree that our checks are good,.
It is possible that there could be a mistake in *both* systems (i.e Prophet and Maverick)
The spreading of knowledge is good. The way we work definitely allows learning.
We are proud of our product knowledge.
Is it really necessary for things to balance to the last cent? We should reduce our quality standards.
Because of the good spread of knowledge, latter stages of the project should go quickly.

Testing bottleneck, version control, time management
The prophet core will be one hundred percent by the time I leave.
Jeremy will have the core right by the time he leaves.
I am worried about version control when Jeremy goes.
There could be a synchronisation problem - we need to design a version control system for test data and the Prophet core.
MAP is a bottleneck - you can not test without him. It is not easy to get access to MAP.
We need to compartmentalise our day so we can get access to people.
Emphasis on quality retards progress.
I am now testing a product I put on two months ago - we need to shorten the testing cycle.
MAP's time needs to be managed better.
Other people should be trained to test on Maverick.
Everyone has to rely on MAP for testing, and testing takes 70% of the time!
You could get done just as quickly with a two man team because testing is the bottleneck not implementation.

We need another tester. Implementing takes 30% of the time and testing takes 70%, yet we have four implementers and only one tester.
The bottleneck will get worse with time.
There is a big backlog of testing.

Recognition
The team leader (John) is not really on the project so how can he judge our work or progress?
We need an effective way for management to recognise our achievements.
Management have no insight into what we do.
We should have a feedback strategy - it should involve group recognition.
We need quicker, more immediate feedback.
There should be regular progress measurement meetings that also sets expectations.
There should be both a carrot and a stick.
We need a daily reportback structure.
We need to wrap up old work.
I did a business case and was proud of it.

F2a


Info Browser
BRAIN
Product Knowledge
ORDER ACCORDING TO PLAN IN PROPHET BUSINESS CASE
EXPECTATIONS
Decide on date by which an agreed number of products is to be completed
Assume "no news is good news"
Feedback to sponsors if deadline seems unattainable
PROPHET Knowledge
PROPHET
(GPM, Vol, EV)
Business Knowledge
VB, Excel CashFlow model, Maverick
ACTIVE learning
Weekly Training (particularly new team members)
new knowledge as a result of testing
Business Logic Manual
Retained knowledge, unambiguous record of how we model our business

Prophet Project Design

Sections / Roles

Code Product

Maintenance of Core

Input Parameters & Tables

Documentation
- Business Logic
- Implementation / Product

Quality Control
- Version Control
- Testing Protocol
- Maintainability

Policy Data Provision
- Testing
- Production Runs

Output Format & Variables

Product Implementation Schedule

Project Management

Reviews

Review Process
- Peer Reviews
- Product "Walk-Through"
- Manager Approval

Review Trigger Event
- Product Implemented

Evaluation Criteria

Other

Roles & Responsibilities

Communication

Working Teams / Structure

Resources

Product Implementation Process (Check List)

Code Product
- Actuarial Spec
- Update Input/Core Variables

Input Parameters & Tables

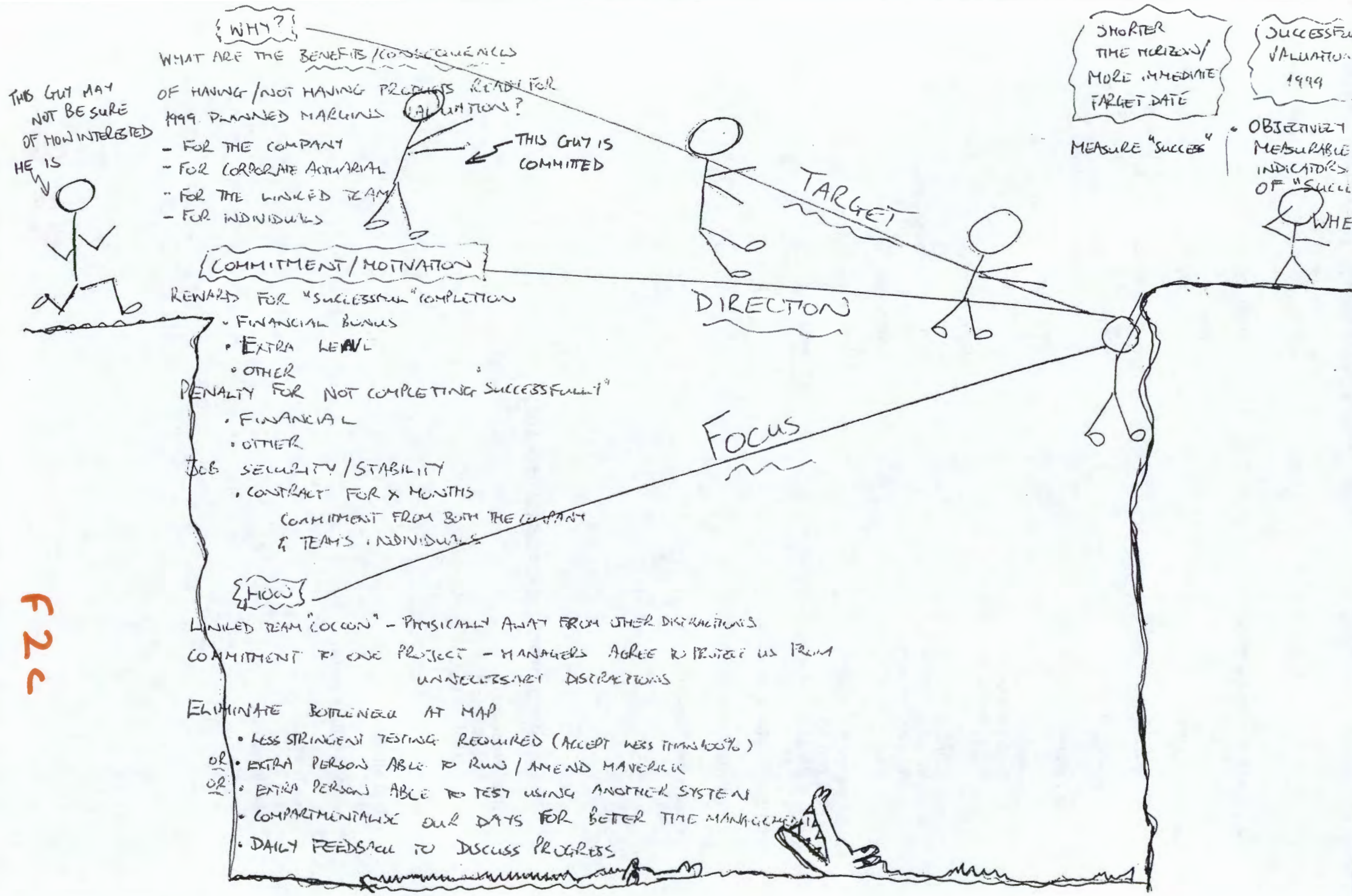
Test Model

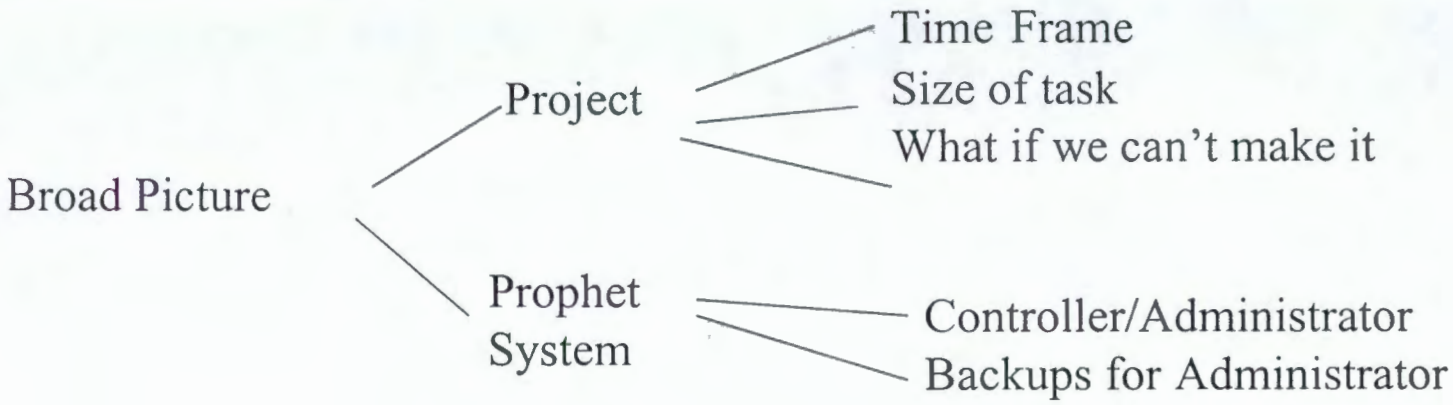
Limited & Bulk Testing

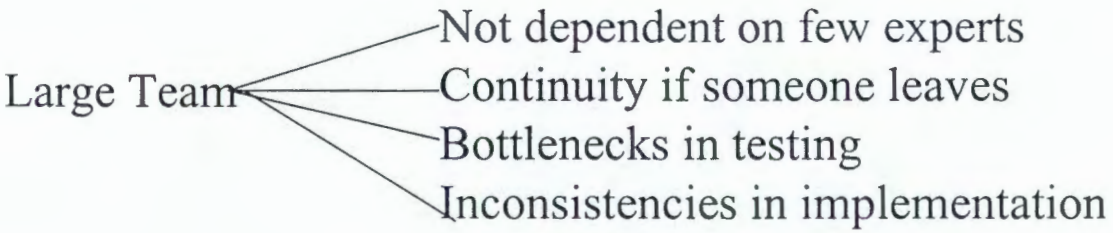
Version Control

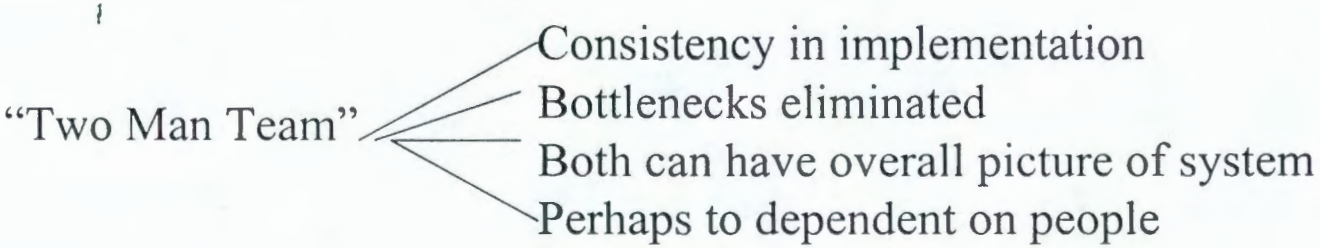
Documentation

F26


{WHY?}
WHAT ARE THE BENEFITS/CONSEQUENCES
OF HAVING/NOT HAVING PRODUCTS READY FOR
1999 PLANNED MARGINS (ALUATION?
- FOR THE COMPANY
- FOR CORPORATE ACTUARIAL
- FOR THE LINKED TEAM
- FOR INDIVIDUALS
THIS GUY MAY NOT BE SURE OF HOW INTERESTED HE IS
THIS GUY IS COMMITTED
SHORTER TIME HORIZON/ MORE IMMEDIATE TARGET DATE
SUCCESSFU VALUATION 1999
MEASURE "SUCCESS"
• OBJECTIVELY MEASURABLE INDICATORS OF "SUCCE
WHE
TARGET
DIRECTION
{COMMITMENT/MOTIVATION}
REWARD FOR "SUCCESSFUL" COMPLETION
• FINANCIAL BONUS
• EXTRA LEAVE
• OTHER
PENALTY FOR NOT COMPLETING SUCCESSFULLY
• FINANCIAL
• OTHER
JOB SECURITY/STABILITY
• CONTRACT FOR X MONTHS
COMMITMENT FROM BOTH THE COMPANY
& TEAMS / INDIVIDUALS
FOCUS
{HOW}
LINKED TEAM "COCOON" - PHYSICALLY AWAY FROM OTHER DISTRACTIONS
COMMITMENT TO ONE PROJECT - MANAGERS AGREE TO PROTECT US FROM
UNNECESSARY DISTRACTIONS
ELIMINATE BOTTLENECK AT MAP
• LESS STRINGENT TESTING REQUIRED (ACCEPT LESS THAN 100%)
OR • EXTRA PERSON ABLE TO RUN / AMEND MAVERICK
OR • EXTRA PERSON ABLE TO TEST USING ANOTHER SYSTEM
• COMPARTMENTALIZE OUR DAYS FOR BETTER TIME MANAGEMENT
• DAILY FEEDBACK TO DISCUSS PROGRESS
F2C


Broad Picture
Project
Time Frame
Size of task
What if we can't make it
Prophet System
Controller/Administrator
Backups for Administrator


Large Team
Not dependent on few experts
Continuity if someone leaves
Bottlenecks in testing
Inconsistencies in implementation


"Two Man Team"
Consistency in implementation
Bottlenecks eliminated
Both can have overall picture of system
Perhaps to dependent on people

F2d

Jeremy Spina

Resources:

Name	Role	Medium	Proficiency
Map	Tester	Maverick	3
Jeremy	Implementor	Prophet	3
Mark	Implementor	Prophet	3
	Tester	VisBasic	3
Bjorn	Implementor	Prophet	3
	Tester	Excel	3
Malcolm	Implementor	Prophet	3
	Tester	Excel	2
John	Implementor	Prophet	1
	Tester	Excel	1
Manfred	Implementor	Prophet	1
	Tester	Excel	1

Proficiency indicates the familiarity/comfort of working with the particular interface
1 = New user
2 = Reasonable familiarty
3 = Proficient user

Description of media

Name	Intent	Processes	Products on	Capacity
Prophet	Ultimate system	QPM, Valn, Embedded Value	First Link, EEP, Master Adaptor, FDB, EPGL	Full
Maverick	Main current system, used in testing Prophet	QPM, Valn, EV	SLAL prods, except EEP, FDB, EPGL	Full
VisBasic *Note 1*	Testing Prophet products not on Maverick	QPM, Valn	Master Adaptor, FDB, EPGL	Full
Excel *Note 2*	Testing Prophet, familiarizing with cash-flow model	QPM	Master Adaptor, some variations	Indiv. Model points

Note 1

- If used for future testing, will take time to implement products on VisBasic system.
- Not guaranteed to tie up with Maverick
- Cannot currently handle Embedded Value (EV)

Note 2

- Not a suitable medium for testing more than a very limited number of model points
- Mainly used as a learning tool – to understand cash-flow process
- John is currently designing own cash-flow model from scratch

F2e

Problems

Difficulty of time forecasting

Deadlines set by negotiation

But often need to predict amount of time which a process will take, where person has never undertaken process before.

Results in (most often) people underestimating time required.

90/10 principle: where 90% of the work takes 10% of the time, and the remaining 10% takes 90% of the time. Hence, difficulty measuring progress in a meaningful way

Bottleneck in testing phase

Problem that Map is realistically the only person testing implementation of products via an independent system (Maverick)

Map has huge demands on his time anyway.

Add to this the fact that testing (where Map is involved) takes 70% of the time of the whole process. Map has to cater to 4 implementors all making demands on him to help them test the implementation of their products on Prophet

BIGGEST PROBLEM (as I see it).

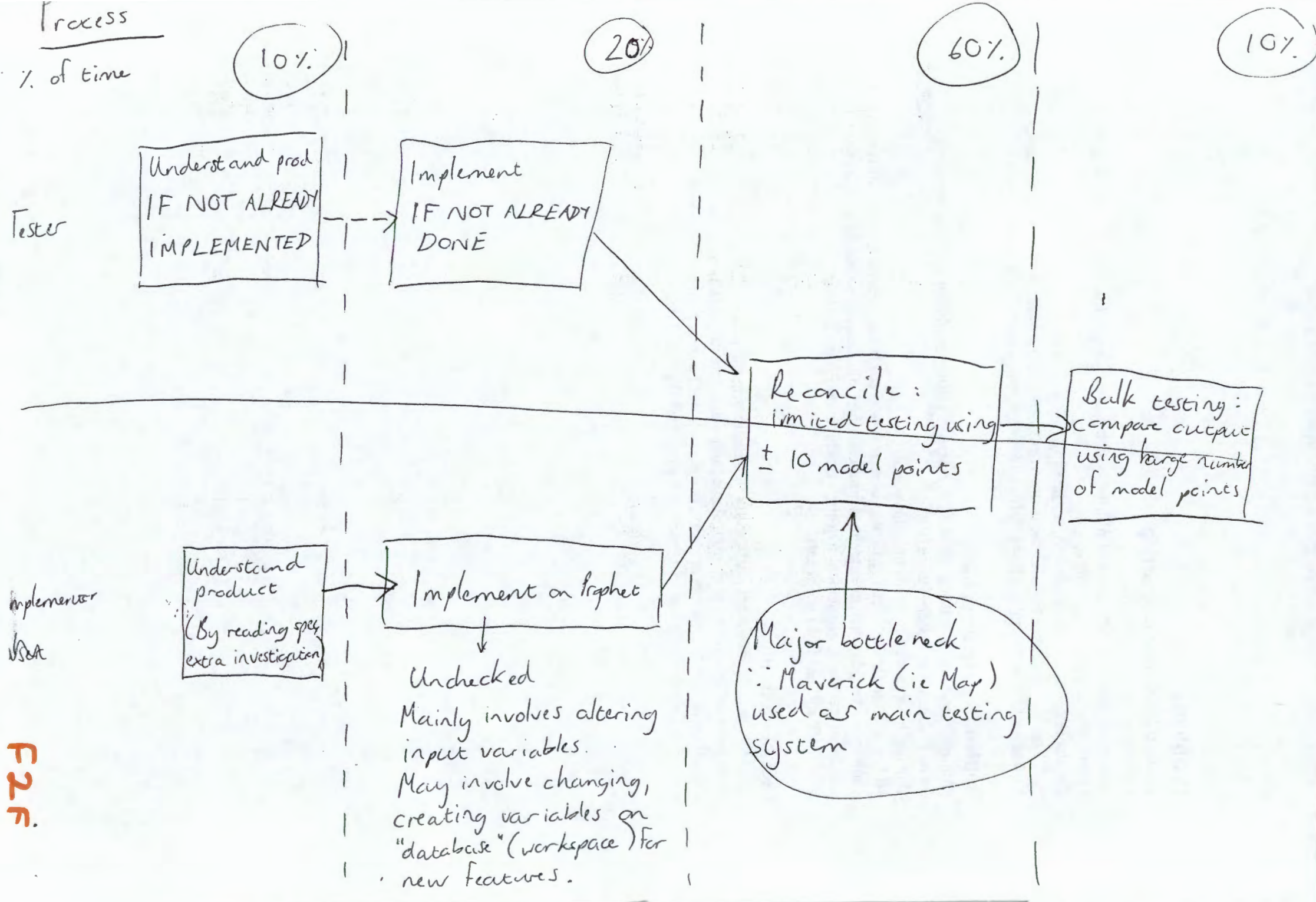
Lack of motivation:

People haven't "committed" sufficiently to deadlines in the past

Generally size up amount of time available, and make sure all work is done by then.

If deadline is seen to be slipping away, people will not go out of their way to meet it – as long as they feel they have time to catch up before the "final deliverable"

F2e


Process
% of time
10%
20%
60%
10%
Tester
Understand prod IF NOT ALREADY IMPLEMENTED
Implement IF NOT ALREADY DONE
Reconcile: limited testing using ± 10 model points
Bulk testing: compare output using large number of model points
Implementer
VBA
Understand product (By reading spec, extra investigation)
Implement on Prophet
Major bottleneck ∴ Maverick (ie May) used as main testing system
Unchecked
Mainly involves altering input variables.
May involve changing, creating variables on "database" (workspace) for new features.
F2F.

(1) MAP

Prophet Project Roles & Design

OLD PRODUCTS

Project Management — **John (SLAL & Medhelp) & ~~Ruguru~~ (Other)**

Oversees the following functions:
Integration Management – ensuring that the various elements of the project are properly co-ordinated (project plan development & execution, and overall change control).
Scope Management – ensuring that the project includes all the work required to complete the project successfully.
Time Management – ensuring the timely completion of the project (activity sequencing, duration estimation, and schedule development & control).
Quality Management – ensuring that the project satisfies the needs for which it was undertaken.
Resource Management – ensuring that appropriate resources are available (human & equipment) and making the most effective use of them. (organisational planning and manage resource conflicts with other projects).
Risk Management – the processes concerned with identifying, analyzing and responding to project risk.

SS IMPLEMENT

Implement Products — **Team**

Code Products – obtain the actuarial spec, and update Prophet input and core variables to cater for QPM, valuation (planned margins) and embedded value (Agree core changes with manager of core).
Input Parameters & Tables – ensure the parameter files and tables are created and correct (Agree additional data requirements with data manager).
Documentation – document the implementation of a product on Prophet, and ensure consistency with the business logic manual.

Management of Prophet Core — **Mark (SLAL) & Trevor (Other)**
Bjorn (Medhelp)

Maintain Core Variables – ensuring the integrity of the core & implement / import changes to the core.
Output Variables & Format
Documentation of Core
Global Parameters
Prophet Versions – upgrade databases to later versions once available.

EDUCATION - CORE MANAGEMENT (BACK UP)

Quality Control — **Map (SLAL) & ~~???~~ (Other)**

& VERSION CONTROL

Version Control – ensure that the product files, master database, tables, parameter files, test data and documentation are updated / changed in a controlled fashion.
Testing Protocol Management – ensuring adherence to the testing protocol.
Limited & Bulk Testing – oversee and ensure the testing of all products implemented onto Prophet.
Test Model – develop or update the test model.

AT THIS

Communications Management — **Mark (SLAL) & Ruguru (Other)**

PROJECT DRIVER:
ASPEN SS (4)

+ (OUTSOURCE)

F3a

PM

Communication Management – ensure the timely and appropriate generation, collection, dissemination, storage and ultimate disposition of project information.
Business Logic Manual
Product & Core Documentation Management – ensure the necessary documentation is carried out.

Data Management **Map (SLAL ~~& Other) & Bjorn (Medhelp)~~**

Bulk Testing & Production Policy Data – specify the data fields and selection criteria for policy data, and co-ordinate the provision of data from IS.
Results Reporting – ensuring that the necessary result information is easily accessible for reporting.

EXIA-1 3 F (4)

Team Development **Mark**

Training
Motivation

Reviews

Conducted on "completion" of a product (implemented, tested & documented).
Product "walk-through"
Assessment criteria:
- Accuracy
- Speed
- Documentation
- Spreading of knowledge
- Team work / cooperation

Peer review with feedback to line manager.

Product Implementation Schedule

Duration Estimation – assessed by team members (subject to the approval of the project manager).
Allocation of Resources – team decides which products are allocate to which person (subject to the approval of the project manager).
Reviewed monthly.

Testing Protocol

1. Implement Product
2. Limited Testing
3. Bulk Testing
4. Import Product
5. Bulk Test Core (if there were changes to core)
6. Bulk Test Product on Master Database
7. Update Master Copy
8. Archive Test Data & Results

F3a

Documentation

Business Logic Manual
Core Documentation
Schedule (dates, implementer & tester)
Business Case
Product Documentation (simplifying assumptions & peculiarities)
Input Assumptions

F3a

(2)

Role	Description	Comments	Proposed Person
Project Leader/Driver	Responsible for ensuring that the team remains focused on and achieves the final objective; • With a good understanding of the broader picture and how to get to the objective • Ensure adequate resources available for the project • Resolve emerging conflicts between/among team members and those external to the team • Have skills to deal with crisis that may emerge from time to time • Must be sensitive to the needs of the team members including sound training, study leave, and at times when one just has a bad day • Must be creative in ways that keeps the team motivated throughout the project • Draw on the strength of individual members by delegating those tasks that that individual can add greatest value • Must not only be fair but must be seen to be fair	From what I have observed within the current Prophet Project Team, I see all the members being strong in some of these areas but weak in others. I would therefore recommend that we consider someone outside of the immediate team, to become team leader	Julian Day???
Principal Technical Expert	• Ideally with the highest level of technical knowledge and understanding of the task ahead. • Very good understanding of the system and the products that must be implemented • Can handle the more difficult tasks within the project • Should have a very good feel of the skills base in the team • Hence in a good position to assist the Team Leader in allocating specific tasks • Would be expected to gradually pass on his/her skills to one or more team members • Ideally should be an easily approachable person	The person would need to have worked with Prophet and on products for a while to develop the necessary skills	Jeremy
Principal Trainer	• Ideally with good level of technical knowledge and understanding of the task ahead. • Good understanding of the system and the products that must be implemented • Should have a very good feel of the skills base in the team • Good in communication skills to enable him train his/her colleagues • Should be patient and approachable especially by new members of the team • Will be in a good position to assist the Team Leader and Principal Technical Expert -in allocating specific tasks -setting shorter term deadlines since they have a good feel of what it takes to accomplish each task	The person would need to have worked with Prophet and on products for a while to develop the necessary skills Keen to do the job	Mark
Tester (s)	• Responsible for quality control	Ratio of implementers to testers should be improved	MAP & John
Implementers	Responsible for ensuring that: • All products are fully implemented onto Prophet and within the agreed deadlines as much as possible • Ensure quality work done to meet the objective at the		Bjorn, Malcolm, Manfred, Principal

F36

	highest possible standards		Expert and Trainer
Hangers On	Responsible for those tasks that seem peripheral to the project but nevertheless are an important part of the whole process. In particular: • to as far as possible provide working technical product specifications of the AAL and GLAC products • tasks should be included in project planning and ideally deadlines should be agreed here as well		Ruguru
Team Manager	• Responsible for the human resource aspect of the team and working with other leaders and other members of the team, deal with: -Leave plans -Promotions/salary increases • Maintain a balance between other demands on the team and the Prophet project • Be a link between the Prophet Team and senior management	Should be a senior member of the team in terms of current grades Has worked with team and has a sense of what each member is capable of Develop a way of assessing performance and how to reward high achievements	John
The Team	Responsible for delivering the Planned Margins Valuation results by June 1999. • Focus on the objective and learn to work with each other with due regard to our individual qualities and faults. • Develop a system which we can use for peer reviews		ALL

F36

PROJECT MANAGER: John should be project manager, driver (ie responsible for coordinating the project activities, setting the deadlines and delegating products) and a tester having the product knowledge and experience (though less exposure to Prophet). He should also attend to "soft" issues – in fact this does not in my mind significantly alter his present position. In this way he is directly involved with an aspect of the project (and is also directly bearing some but not all, of the difficulties) and gains insight into individual performance as he works with team members to test products. This would give an added insight into performance appraisal.
The technical quality of the work should be evaluated by the team as a whole – they rate the difficulty of a particular product and a rough estimate of the completion time (this would be done when planning and should be a moving target that is adjusted as need be according to a weekly feedback session with the team).
TRAINER & IMPLEMENTER: Mark has a natural bent towards training and should be the principal question answerer with respect to Prophet. His role should be to guide Ruguru, Malcolm and myself (to a lesser extent) and to continue implementing products as usual.
TESTING: An initial simple test can be set up on a spreadsheet by MAP with more detailed testing to be done with MAP or John. This balances the need for test independence and releasing the strain on the tester.
COMMENTS: It is important that the valuation deadline be broken down from an overwhelming single target to smaller and more attainable goals. How about having mini valuation deadlines along the way with some reward as well? This would go some way to reducing worry that the project is not on target by building confidence that the team can meet smaller ones. (Exact dates would have to be negotiated particularly with reference to Exams).

(4) SEN...1

Roles

Role	Description	People
Project manager	• Keep track of everyone's activities • Monitor overall progress of project • Help allocate time demands made by other projects	JF
Project Driver	• Deal with "soft issues" • Sort out disagreeements / conflicts between people if they arise • Act as motivator on an overall level, and at an individual level, if need be • Make sure project progresses smoothly • Seek external help to overcome obstacles that may arise *Should probably be combined with Project Manager role*	JF??
Implementor	• Implement products on Prophet • Independent tester, using output from Maverick, as far as is possible • Final reconcilliation, in concert with Tester	MB MW MMa BW
Tester	• Use Maverick to generate output for initial testing of results by Implementor • Final reconcilliation, in concert with Implementor	MMe JF
Trainer	• Train new users on how to use Prophet • Answer questions / handle difficulties which arise in the use of Prophet on day-to-day basis	MB
Prophet Maintainer	• Maintain Master Version of unit-linked database (workspace) • Co-ordinate version control, alterations	MB

Notes:

1. JF's responsibilities as Manager/Driver will naturally restrict the amount of time he is able to spend on testing. In addition, need to take account (at least initially) of the fact that he will be less familiar with Maverick. I suggest assigning one quarter of the testing responsibilities to John, and three-quarters to Map.
2. In assessing demands on MB's time as an Implementor, need to take account of extra responsibilities as Trainer / Maintainer.

Reviews

- Responsibilities and deadlines agreed by negotiation
- Once agreed, reponsibility for sub-tasks devolves to individuals involved
- Peer review of progress
- Driver consults with *individuals* on progress on a weekly basis
- If the individual so desires, they may arrange to discuss progress/difficulties with the driver more frequently.
- Team discussion involving all individuals on a monthly basis (avoids too-frequent meetings involving whole team when their presence is not required for a large part of the meeting)

F3d

F3c

(5) Malcolm

Adrian Burke

From: Malcolm Weir
To: Adrian Burke
Subject: Prophet Planning
Date: 06 July 1998 5:30PM

My feelings are :

John fills the role of "project manager" and "project driver " but becomes involved in the technical side of things by committing to become involved in the testing side of things ,as discussed.

I see this role including the overall management of the project in terms of co-ordinating meetings monitoring progress of the project in relation to deadlines, as well as communicating with higher levels of management. It would also include the softer issues of management. If time allows, then he can also be involved in implementation, but no hard commitment need be made on this aspect.

Reviews should be conducted by the manager, with significant input from the team members (The degree to which this differs from the current situation depends on how much feedback people have given to management in the past at times of PPM reviews)

For testing bottleneck issue :

I liked both suggestions put forward yesterday, and so suggest that we combine them. i.e. Let John become involved in testing by becoming familiar with Maverick. (As noted above, this also allows John's involvement on a technical side). Also let people be given Maverick output with which to do initial testing.

Other issues : I noticed on someone's notes yesterday the idea of regular feedback sessions to pass on new knowledge gained through implementation/testing to other team members. This would be very beneficial for our learning.

F3e

(6) SPA

Roles in the Prophet Project

This note covers the roles that I see as being directly part of the Prophet project. It also covers one other role which is important to the success of Prophet but which I did not see as part of project management of Prophet itself.

For each of the roles, I have outlined what I see the role consisting of as well as the my recommended candidates.

1. Project Management

- Ensuring that all people involved in the project are kept informed with relevant information. What is "relevant" to people will depend on their role as well as their "interest" in the big picture.
- Motivating the team.
- Project timetable - keeping track of progress and updating
- Ensure that required tasks are appropriately allocated (many factors affecting "appropriate allocation"). This allocation itself need not be done by the project manager - but he should be happy with it.
- Ensuring that project sponsor (Ady) is informed and happy with progress.

Candidates : -1 person required
- Adrian, John

2. Implementation of Products on Prophet

- Implementing products on Prophet (!).
- Testing the implementation
- Communicating any "extra" or changed requirements for data to the person responsible for specifying data requirements. (See the role 4.). These changes will arise as people find out peculiarities of products as they start implementing them. I expect that this could happen quite a bit with old AAL and GLAC products.

Candidates : - any number of people
- Björn, Malcolm, Manfred, Mark, possibly John, possibly Ruguru (ex AAL/GLAC/SLA products?)

3. Prophet Technical Management (1 person)

- This role involves managing the technical side of Prophet including version control. Generally ensuring that Prophet as a system is properly maintained from technical point of view.
- Would expect the person filling this role to have (or to build up) greater knowledge of how Prophet works than the implementors.

Candidates : - 1 person required (although useful if the knowledge is spread)
- Mark, Malcolm, Björn
- Mark probably has the best technical knowledge to fulfill this role, although I imagine that Malcolm could pick up this knowledge pretty quickly. Björn would also be able to fulfill this role, although I'm not sure he is best suited to it.

F3F

4. Implementation of PC database and Specifying Data Requirements

- Specifying what input data is required from mainframe and the form of the download files.
- Liaising with "Mainframe Download Project Manager" regarding requirements (i.e. is what required feasible, what are other possibilities, etc.)
- Specifying what output data is required from Prophet and its form.
- Implementing a PC database to store the download data as well as the results data.
- Liaising with Prophet Product implementors regarding data requirements

Candidates : - 1 person required
- Map

5. Independent Tester

- Providing an independently implemented system against which Prophet can be tested.
- I think that this role needs to be looked at pragmatically : it might not be practical for all products.

Candidates : - Map for most SLAL products, Mark for some others, ?? for old AAL/GLAC/SLA products (if this role is needed for them)

6. Training New Users on Prophet

- Train new users how to use the Prophet package
- Train new users in product knowledge, including how we model aspects of the products design. There is possibly not much initial formal training required here as people tend to pick up this knowledge along the way. However the "trainer" would be the first person a new user would refer to when they have questions.
- Keeping the business logic manual up to date (possibly).
- I don't see this as a full time role (although it could fully occupy a person at certain times). .
- This person would also be involved in implementing products. Would need to decide carefully how much time ought to be spent on training as there could be a danger of spending too much time on training.

Candidates : - 1 person required but the role could be shared/rotated
- Mark, anyone else in the team
- I think Mark would be best in this role.
- I don't think that the person/people who take on this roles should also take on both the implementing and technical expert roles as well as the training role all at once. (But it might be feasible.)

7. Implementation of Downloads of Data off the mainframe (outside direct scope of the Prophet Project)

- This involves the actual programming of the mainframe downloads.
- Agreeing timetable of implementation with the Prophet Team.
- Keep the "PC database Implementor" (ugh!) informed of progress of these downloads.

John Ferguson
6 July 1998

F3F

The Linked Team PROPHET Project

by Björn

Project Manager : Handles "soft issues"* time management (a coordinating role)
* motivation of team, concentrating on maintaining direction
* organising feedback sessions

Implementers : Implement and test individual products on PROPHET
Testing is done by each individual (and MAP where necessary) using MAP's suggested spreadsheet as a basis

Team : Manage each other and themselves
Justify their (lack of) progress to each other
Ask/provide help to each other
Provide each other with technical expertise so that each implementer can fulfill his role properly

Globetrotter : Travel around the world
Send postcards to the team from exotic locations all over

I'm not fussed who takes which roles. A suggestion is :
Project manager : Mark or John
Implementers : the rest of us except Jeremy (see globetrotter)
Team : all of us
Globetrotter : Jeremy

I think that MAP's suggestion for eliminating the testing bottleneck, ie the spreadsheet of MAVERICK model point results to be used by individual implementers as their basis for testing, could work. Ultimately though, he will still be the one who has to "OK" the accuracy of each product on PROPHET. I'm not sure that this is ideal. Ideally we would have more than one person involved in testing

I think it is important that THE TEAM is committed to the same goals. It is no good if the project manager is concerned about one success criterion (eg time) while the implementers are worried about another (eg accuracy of spreading of knowledge). How do we ensure that we work as a team?

Some people prefer the carrot, others prefer the stick. It is the project manager's task to handle this. For me personally the stick tends to be more effective. I would be driven/committed to do my part in accomplishing our goal for fear of letting the team down. The encouragement I'd need would be seeing the commitment of others in the team to accomplishing our goal.

F3g

PROPHET TEAM PROPOSAL

Mark Billam - 7 July 1998

The following topics are discussed in the PROPHET Business case:

- Scope of project (although this is now expanded to include AAL, SLA and GLAC).
- Risks.
- Critical success factors.
- Which products need to be implemented and in which order. The reasoning behind the order in which the products should be implemented is discussed in the document.

I have described below, briefly, how I see each person fitting into the project. In the light of his departure, I have not mentioned Jeremy. I see his role as distributing his responsibilities to other team members during July 1998.

Ady

- Project sponsor
  - Will do anything to ensure that the project succeeds!

John

- Schedule work
  - Schedule (with team input) the flow of work and ensure that the team is happy with and aware of deadlines.
- Quality management
  - Ensure that coding, testing and documentation are up to our agreed standard.
- Resource management
  - Ensure that the necessary resources (people, skills, hardware, software) are available to successfully complete the project.
  - Monitor that resources are optimally used.
  - Assign resources to other projects.

Mark

- Core maintenance
  - Maintain PROPHET core when required by team.
  - Document PROPHET core.
- Team development
  - Learning / training.
  - Motivation, enthusiasm, necessary skills.
- Documentation
  - Business Logic Manual.
  - Core documentation (built into Business Logic Manual).
  - Product documentation management (collate product documentation done by each team member).

Map

- Data management
  - Provide policy data.
  - Store results.

- Quality management
  - Provide Maverick output for QPM, valuation and EV model point testing.
  - Provide Maverick output for QPM, valuation and EV bulk testing.
  - Archive test data and results after a product is completed (these results may be needed later if changes have been made to the core).

Björn, Malcolm, Manfred

- Product programming
  - Understand product spec.
  - Code product on PROPHET.
  - Ensure decrement, global and parameter tables are correctly set up.
  - Test the product against Maverick (or VB prg if necessary).
  - Complete product documentation and pass on to Mark to add to the product manual.

Björn

- Version control
  - Allow the release of the PROPHET workspace from Mark's PC onto PROPMAST. A separate user ("prophetmaster") should be set up so that a deliberate action is needed before anything can be copied onto PROPMAST.
  - Ensure that core documentation has been updated before allowing a release onto PROPMAST.

F 3 h

Prophet Project Roles & Design

Project Management — **John (SLAL, Medhelp, Old linked products)**

Oversees the following functions:
Integration Management – ensuring that the various elements of the project are properly co-ordinated (project plan development & execution, and overall change control).
Scope Management – ensuring that the project includes all the work required to complete the project successfully.
Time Management – ensuring the timely completion of the project (activity sequencing, duration estimation, and schedule development & control).
Quality Management – ensuring that the project satisfies the needs for which it was undertaken.
Resource Management – ensuring that appropriate resources are available (human & equipment) and making the most effective use of them. (organisational planning and manage resource conflicts with other projects).
Risk Management – the processes concerned with identifying, analyzing and responding to project risk.
Communication Management – ensure the timely and appropriate generation, collection, dissemination, storage and ultimate disposition of project information.

Project Driver

Deal with soft issues.
Sort out disagreements and conflicts between people if they arise
Act as motivator on an overall level and at an individual level if need be
Ensure the project progresses smoothly.
Seek external help to overcome obstacles that may arise

Implement Products — **Bjorn, Malcolm, Manfred, Mark**

Code Products – obtain the actuarial spec, and update Prophet input and core variables to cater for QPM, valuation (planned margins) and embedded value (Agree core changes with manager of core).
Input Parameters & Tables – ensure the parameter files and tables are created and correct (Agree additional data requirements with data manager).
Documentation – document the implementation of a product on Prophet, and ensure consistency with the business logic manual.

Management of Prophet Core — **Mark (SLAL)**

Maintain Core Variables – ensuring the integrity of the core & implement / import changes to the core.
Output Variables & Format
Documentation of Core
Global Parameters
Prophet Versions – upgrade databases to later versions once available.
Educate others how to perform core management, backup

Quality Control — **Map**

Version Control – ensure that the product files, master database, tables, parameter files, test data and documentation are updated / changed in a controlled fashion.
Testing Protocol Management – ensuring adherence to the testing protocol.

F4

Limited & Bulk Testing – oversee and ensure the testing of all products implemented onto Prophet.
Test Model – develop or update the test model.

Data Management **Map (SLAL & Other)**

Bulk Testing & Production Policy Data – specify the data fields and selection criteria for policy data, and co-ordinate the provision of data from IS.
Specifying the output data required from Prophet and its form.
Results Reporting – ensuring that the necessary result information is easily accessible for reporting.
Liaising with "Mainframe Download Project Manager" regarding requirements (i.e. is what required feasible, what are other possibilities, etc.)
Implementing a PC database to store the download data as well as the results data.
Liaising with Prophet product implementors regarding data requirements

Team Development **Mark**

Training
Motivation
Business Logic Manual
Product & Core Documentation Management – ensure the necessary documentation is carried out.

The following is a list of issues or areas that need to be agreed within the team.

Reviews

Conducted on "completion" of a product (implemented, tested & documented).
Product "walk-through"
Assessment criteria:
- Accuracy
- Speed
- Documentation
- Spreading of knowledge
- Team work / cooperation

Peer review with feedback to line manager.

Product Implementation Schedule

Duration Estimation – assessed by team members (subject to the approval of the project manager).
Allocation of Resources – team decides which products are allocate to which person (subject to the approval of the project manager).
Reviewed monthly.

Version Control

Testing Protocol

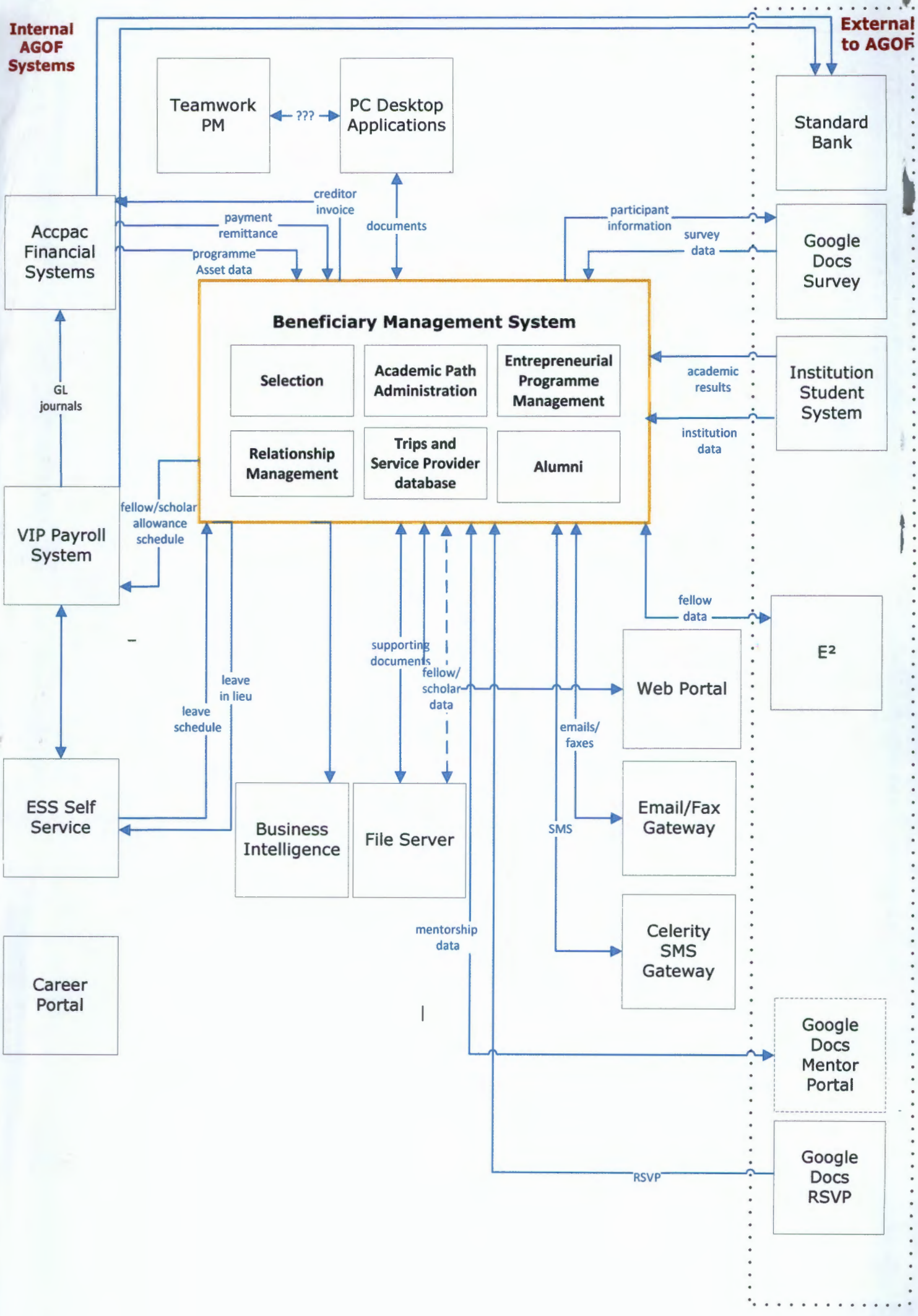
1. Implement Product
2. Limited Testing

3. Bulk Testing
4. Import Product
5. Bulk Test Core (if there were changes to core)
6. Bulk Test Product on Master Database
7. Update Master Copy
8. Archive Test Data & Results

Documentation

Business Logic Manual
Core Documentation
Schedule (dates, implementor & tester)
Business Case
Product Documentation (simplifying assumptions & peculiarities)
Input Assumptions

F4


Internal AGOF Systems
External to AGOF
Teamwork PM
???
PC Desktop Applications
Standard Bank
creditor invoice
payment remittance
programme Asset data
documents
participant information
survey data
Accpac Financial Systems
Google Docs Survey
Beneficiary Management System
Selection
Academic Path Administration
Entrepreneurial Programme Management
Relationship Management
Trips and Service Provider database
Alumni
academic results
institution data
Institution Student System
GL journals
fellow/scholar allowance schedule
VIP Payroll System
fellow data
E²
supporting documents
fellow/ scholar data
Web Portal
leave in lieu
leave schedule
emails/ faxes
ESS Self Service
Business Intelligence
File Server
SMS
Email/Fax Gateway
Celerity SMS Gateway
mentorship data
Career Portal
Google Docs Mentor Portal
Google Docs RSVP
RSVP

Purpose of thesis.

Emphasize usefulness to the