

# **Developing a Methodological Generic Framework through Introducing Autonomy and Self-Adaptation to Information Systems Thinking**

**Peter Alasdair Kinloch**

A thesis submitted in partial fulfillment of the requirements  
of Liverpool John Moores University for the degree of  
Doctor of Philosophy  
of the  
Liverpool John Moores University  
School of Computing and Mathematical Science

December 2008

**BEST COPY**

**AVAILABLE**

*To my family*

## **Abstract**

There is a requirement for systems methodologies and approaches that can cope with real life information systems that are subject to changing situations and therefore changing requirements. This has not been achieved previously and has seen a gap open up between information systems and information technology. It is recognised that information technology solutions can adapt to changing situations and subsequently changing requirements, however, this has not been possible for information systems thinking. This represents itself in the real world through information systems being used that no longer meet their original objectives and can provide a significant blockage to achieving effective work.

To achieve this a generic framework has been developed that incorporates approaches of both hard and soft systems thinking, which provides self-adaptive and autonomic capabilities. The generic framework is based upon the Soft Systems Methodology structure and is extended to include and accommodate the additional functionality brought by Viable Systems Model. By utilising these two approaches into the developed generic framework functionality is provided that is capable of flexibility and adaptation as requirements change.

The generic framework has been tested in a case study involving a large organisation whose current information systems were not providing staff with all the appropriate information they needed to work effectively. This led to both poor performance and an inability to change to situations as they arose. The application of the generic framework led to the redesigning of major information systems within the organisation and saw a significant improvement in performance.



## **Acknowledgements**

My deepest thanks and gratitude goes to Dr. Michael Francis of the School of Computing and Mathematical Sciences, Liverpool John Moores University for his invaluable supervision, constant help and support, constructive feedback and his incredible ability to listen to my half-baked ideas without laughing.

I would also like to express my sincere appreciation to Mrs. Hülya Francis, Professor Azzelarabe Taleb-Bendiab, Dr. Mark Taylor and Mr. Andy Laws for their guidance, assistance and constant willingness to enter into discussions in relation to this study.

I would like to thank the academic staff, administrative staff and technicians in the School of Computing and Mathematical Science, Liverpool John Moores University for their support.

My colleagues at Merseyside Police should also be praised for their assistance, in particular Steve Smith, Dave Root, Paul McBride, Stefan Luszniak, Nick Mills, Sarah Farrell, Alan Wood and Gary O'Rourke who at various stages provided either study related assistance or moral support.

Finally, I would like to offer my thanks to my parents Peter and Berni, my brother Iain and also Sean who ensured that I was never short of support at the times when I needed it most. You all have a special place in my heart and I dedicate this to you.

Table of Contents

Abstract.....ii

Acknowledgements..... iii

1. Introduction..... 7

1.1 Thesis Introduction ..... 7

1.2 Motivations ..... 7

1.3 Challenges Facing the Development of the Generic Framework ..... 9

1.4 Contributions Made to the Field of Information Systems Thinking by this Research..... 11

1.4.1 Conflict Management..... 12

1.4.2 Control ..... 13

1.4.3 Reconfiguration..... 13

1.4.4 Repair and Adaptation Strategies..... 13

1.5 Thesis Structure and Organisation ..... 14

2. Methodologies Employed..... 16

2.1 Introduction to the Methodologies Employed ..... 16

2.2 Soft Systems Methodology ..... 17

2.3 Viable Systems Model ..... 19

2.4 Prince2 ..... 20

2.5 Effective Technical and Human Implementation of Computer-based Systems (ETHICS)..... 22

2.6 Multiview Framework ..... 23

2.7 Structured Systems Analysis and Design Methodology (SSADM) ..... 24

2.8 Hard and Soft Methodologies ..... 26

2.9 Choosing the Methodologies ..... 27

2.10 Linking SSM and VSM ..... 27

2.11 Chapter Summary ..... 29

3. Background..... 31

3.1 Introduction to the Background ..... 31

3.2 Self-Adaptive Software..... 31

3.2.1 Control Systems Theory ..... 33

3.2.2 Dynamic Planning Systems ..... 34

3.2.3 Self-Aware Systems..... 34

3.2.4 Self-Adaptive Software Compared to Control Systems Theory and the Principles that Underpin the Approaches..... 35

3.2.5 Self-Adaptive Research Directions..... 37

3.2.6 Self-Adaptation and Reflective Architectures ..... 39

3.3 Autonomic Computing..... 39

3.3.1 The Autonomic Computing Architecture ..... 41

3.4 Chapter Summary ..... 42

4. Problem Definition..... 44

4.1 Introduction to Problem Definition..... 44

4.2 The Static Management Approach..... 45

4.2.1 Conflict Resolution ..... 45

4.2.1.1 The Negotiation Strategy ..... 46



4.2.1.2 The Arbitration Strategy .....	50
4.2.1.3 The Voting Strategy .....	51
4.2.1.4 The Independence Strategy .....	51
4.2.1.5 The Four Strategies of Conflict Resolution .....	52
4.2.2 Strategy and Planning .....	53
4.2.3 Exception Handling .....	54
4.2.4 The Deliberative Approach.....	56
4.2.4.1 Extensions to the BDI Model.....	57
4.2.4.2 The Epistemic Deontic Axiologic Model .....	58
4.3 The Dynamic Management Approach .....	59
4.3.1 Dynamic Policy Based Management.....	60
4.3.2 Dynamic Event Based Management.....	63
4.3.3 Dynamic Architecture Based Management .....	66
4.3.4 Autonomic Based Management.....	68
4.4 Chapter Summary .....	70
5. Problem Solution .....	71
5.1 Introduction to the Problem Solution.....	71
5.2 Autonomic Control Requirements .....	72
5.2.1 Detection of Conflicts .....	73
5.2.2 Identification of Conflicts .....	75
5.2.3 Classification of Failure and Conflict .....	75
5.2.4 Conflict Resolution Strategies .....	76
5.2.5 Control Rules .....	77
5.2.6 Reconfiguring the Information System.....	77
5.3 Chapter Summary .....	79
6. System Design.....	81
6.1 Introduction to System Design.....	81
6.2 The Control Architecture .....	81
6.3 The First Strand of the Generic Framework .....	82
6.3.1 Registration Functionality.....	82
6.3.2 Discovery and Monitoring Functionality.....	83
6.4 The Second Strand of the Generic Framework.....	84
6.4.1 Framework Manager and Control Functionality.....	85
6.4.2 Diagnosis Functionality .....	85
6.4.3 Control Rules Functionality .....	86
6.4.3.1 Resources .....	88
6.4.3.2 Rules .....	88
6.4.3.3 Responsibilities .....	89
6.4.3.4 Regulations .....	89
6.4.3.5 Recommendations.....	90
6.5 The Third Strand of the Generic Framework.....	91
6.5.1 Repair Functionality.....	92
6.5.2 Self-Adaptive Functionality.....	93
6.5.3 Repair Strategy Functionality .....	94
6.5.4 Reconfiguration Functionality .....	95
6.6 Chapter Summary .....	96
7. System Development .....	97
7.1 Introduction.....	97
7.2 Building the Generic Framework.....	97

7.2.1 Utilising SSM's Lancaster Model.....	97
7.2.2 Incorporating the Viable Systems Model .....	99
7.2.2.1 Overlapping Functionality .....	99
7.2.2.2 Incorporating Decision Making Functionality.....	101
7.2.2.3 Integrating Self-Adaptive and Autonomic Concepts into the Generic Framework .....	102
7.3 Introducing the Complete Generic Framework .....	103
7.4 Chapter Summary .....	105
<i>8. Experimentation of the Generic Framework .....</i>	<i>107</i>
8.1 Introduction to the Experimentation of the Generic Framework.....	107
8.2 Case Study: Merseyside Police.....	107
8.2 Introducing Merseyside Police .....	108
8.2.1 Total War on Crime .....	110
8.2.2 Total Care for Victims .....	111
8.2.3 Total Professionalism.....	112
8.3 Neighbourhood Policing .....	113
8.3.1 Neighbourhood Policing Through Locally Based Officers .....	113
8.3.2 Serious and Organised Crime .....	114
8.3.3 Citizen Focus .....	115
8.3.4 Support Departments .....	116
8.4 Continuous Improvement.....	117
8.5 Developing an Understanding of the Problem Situation .....	118
8.6 Summary of Introduction to Case Study Organisation .....	120
8.7 Application of Generic Framework to Case Study Organisation .....	121
8.7.1 Generic Framework Step 1 .....	121
8.7.2 Generic Framework Step 2 .....	122
8.7.3 Generic Framework Step 3 .....	127
8.7.4 Generic Framework Step 4 .....	128
8.7.5 Generic Framework Step 5 .....	130
8.7.5.1 Resources .....	130
8.7.5.2 Rules .....	131
8.7.5.3 Responsibilities .....	131
8.7.5.4 Regulations .....	132
8.7.5.5 Recommendations.....	132
8.7.6 Generic Framework Step 6 .....	133
8.7.7 Generic Framework Step 7 .....	134
8.8 The New Information System.....	135
8.8.1 Implementing the New Information System.....	137
8.8.2 Management Information Access .....	138
8.8.3 Report Access .....	142
8.8.4 Full Record Access .....	146
8.9 Further Development of the New Information System.....	148
8.10 Chapter Summary .....	149
<i>9. Testing the Generic Framework .....</i>	<i>150</i>
9.1 Introduction to Testing of the Generic Framework .....	150
9.2 Establishing a Testing Platform .....	150
9.3 Establishing a Benchmark.....	152
9.3.1 Testing from a Quantitative Perspective.....	153
9.3.2 Testing from a Qualitative Perspective.....	153



9.3.3 Reporting on the Experimentation ..... 155

9.4 Practicalities: Implementation of a New Crime Recording System ..... 155

9.5 Developments ..... 157

9.6 Chapter Summary ..... 159

10. Evaluation of Experimentation ..... 160

10.1 Introduction to the Evaluation of Experimentation ..... 160

10.2 The New Information System at Merseyside Police – Initial Evaluation.. 160

10.3 Methodological Approach ..... 165

10.3.1 Appropriateness ..... 165

10.3.2 Objectives ..... 166

10.3.3 Approach..... 167

10.3.4 Requirement for Evaluation..... 167

10.3.5 User Involvement..... 168

10.3.6 Environment..... 168

10.4 Quantitative Evaluation of the Generic Framework ..... 169

10.5 Qualitative Evaluation of the Generic Framework ..... 172

10.6 Discussion ..... 176

10.6.1 Conflict Detection..... 177

10.6.2 Conflict Identification and Classification..... 178

10.6.3 Conflict Resolution Strategies ..... 178

10.6.4 Ability to Reconfigure the Information System..... 178

10.6.5 Control Rules ..... 179

10.7 Chapter Summary ..... 180

11. Conclusions..... 181

11.1 Introduction to Conclusions..... 181

11.2 Motivations and Approach Summary ..... 181

11.3 Contributions..... 184

11.4 Achievements..... 186

11.5 Thesis Summary..... 188

11.6 Discussion ..... 191

11.7 Future Work ..... 191

Glossary ..... 193

Appendix ..... 194

Generic Framework User Guide ..... 195

Business Benefits of Niche ..... 204

Bibliography ..... 217

Books ..... 217

Thesis ..... 217

Conference and Journal Papers..... 217

Websites..... 225

References..... 227

# Table of Figures

Figure 1: Adaptation of the Lancaster Model produced by Checkland .....	19
Figure 2: Adaptation of the Viable System Model .....	20
Figure 3: Adaptation of the Prince2 process model .....	21
Figure 4: Adaptation of the Multiview framework .....	23
Figure 5: Adaptation of the SSADM framework .....	25
Figure 6: Mechanism of self-adaptive software according to Garlan and Taylor's architecture .....	38
Figure 7: Model of the control loop representing the autonomic computing architecture .....	41
Figure 8: The hierarchy pyramid of the autonomic computing technologies .....	41
Figure 9: A simplified version of Klein's Contract-Net Protocol .....	48
Figure 10: The control mechanism of Ephrati and Rosenchein's model for adjustable autonomy.....	50
Figure 11: The generic 'Exception Management Meta-Process' model by Klein and Dellarocas . .....	55
Figure 12: Model which summarises the exception handling process by Klein and Dellarocas . .....	55
Figure 13: Relations between beliefs, goals and intentions as stated by Bratman . ...	57
Figure 14: The relationship between management and monitoring as defined by Nuseibeh <i>et al.</i> .....	74
Figure 15: The relationship between actions, strategies and strategy decision making according to K. Barber <i>et al.</i> .....	76
Figure 16: Espejo's representation of a viable system and the components within it. ....	83
Figure 17: The control rules functionality of VSM as discussed by Espejo , which ensures that the generic framework can fully embrace the concepts of self-adaptation and autonomy. ....	86
Figure 18: Adaptation of the Lancaster Model produced by Checkland, as discussed in Chapter 2 .....	98
Figure 19: Meta level depiction of VSM .....	100
Figure 20: Conceptual Model of SSM and its component parts by Brooks . ....	100
Figure 21: Diagram to show how Conceptual Models and the management functionality of VSM will work together in Step 4 of the Lancaster Model. ....	102
Figure 22: Diagram showing how the new generic framework fits within the Lancaster Model of SSM. ....	105
Figure 23: Breakdown of BCUs across Merseyside Police .....	109
Figure 24: Division structure of Merseyside Police .....	109
Figure 25: High level view of the conceptual model with self-adaptive and autonomic functionality. ....	128
Figure 26: Conceptual model of Merseyside Police case study. ....	129
Figure 27: Screenshot of management information access.....	139
Figure 28: Screenshot of management information access.....	140
Figure 29: Screenshot of management information access.....	140
Figure 30: Screenshot of management information access.....	141
Figure 31: Screenshot of management information access.....	142
Figure 32: Screenshot of management information access.....	143



Figure 33: Screenshot of management information access..... 144

Figure 34: Screenshot of management information access..... 144

Figure 35: Screenshot of management information access..... 145

Figure 36: Screenshot of management information access..... 146

Figure 37: Screenshot of management information access..... 147

Figure 38: Screenshot of management information access..... 147

Figure 39: Screenshot of management information access..... 148

Chart 1: Chart demonstrating the positive effect the implementation of the new approach within Merseyside Police has had on the level of crime. .... 170

Chart 2: Chart demonstrating the drop in average custody airlock times across the Force. .... 171

Chart 3: Chart demonstrating the fall in the number of crimes that require reclassification. This highlights the fact that more crimes are being recorded under the correct initial classification. .... 171

Chart 4: Chart demonstrating results of a data quality audit on crime records. The new approach adopted by Merseyside Police shows a sharp rise in compliance to data quality standards, which significantly enhances the investigative process. .... 172

Chart 5: Chart demonstrating training compliance throughout the phased implementation of the new approach within Merseyside Police. .... 173

Chart 6: Chart demonstrating the rise in percentages of public confidence and satisfaction through surveys conducted with victims of crime. .... 176

# **1. Introduction**

## **1.1 Thesis Introduction**

Information starvation can have a negative impact on the performance of organisations. In particular, large and complex public sector organisations with vast information flows can be the sufferers of such starvation. The provision of inaccurate or irrelevant information significantly hampers decision making processes, which leads to a down turn in performance.

Through developed knowledge of information systems, and through the construction of a new generic framework based upon existing methodological thinking and adopting self-adaptation and autonomic principles (taken from IT solutions to facilitate adapting to changes and decision making), the problem of information starvation can be challenged. This thesis demonstrates the development of a generic framework and its application to the problems being experienced within a large organisation.

## **1.2 Motivations**

Over the past two decades IT solutions have become commonplace in both homes and working life. During the past decade the capabilities of IT systems has increased significantly. This evolution has seen IT systems become more powerful and complex as they attempt to cope with the ever increasing demands placed on them. With these ever increasing demands there is a prerequisite that has arisen for further research into generic approaches. Due to the increased complexity of IT systems within organisations there is a need to develop knowledge and understanding of generic functionality as it is no longer possible to provide a 'one size fits all' approach.

The emergence of autonomic computing and self-adaptive systems has seen the world of IT take a big step forward as monitored by IBM [1]. However, the world



of information systems thinking has, so far, been unable to provide a methodology capable of utilising the concepts of autonomy and self-adaptation. In the real world, this has been demonstrated as information starvation, which defines organisational performance.

Historically, according to IBM, methodological approaches have been applied to problem situations in an ad hoc manner. This can lead to a proactive and reactive dichotomy within a large number of organisations that are suffering with information starvation. In some organisations, such as demonstrated by the Office of Government Commerce (OGC) [2], a particular methodology is to be used exclusively, which may not be best suited for the organisation or the problem situation that it is experiencing.

Whilst IT capabilities have made huge steps forward only limited consideration has been given to cultural, political and philosophical aspects that exist within any organisation as discussed by Checkland [3]. As a consequence, it is not uncommon to introduce IT solutions that either do not resolve the issue or are not popular amongst the staff as they were not consulted during the requirements analysis stage.

Therefore, the research conducted here establishes a generic framework that can be utilised to develop new information systems that take into consideration a cultural, political and philosophical aspect that exist within organisations and that fully embraces concepts of self-adaptation and autonomy.

Laws [4] demonstrates that information systems thinking has produced powerful methodologies that enable analysts to conduct their work and develop appropriate resolutions. However, a methodological approach does not exist that assists the analyst in defining both the 'whats' and the 'hows' of problem situations. In essence, this means that the analyst could either implement a methodology to help them establish what the problem situation is, or how to implement a solution to the problem situation. An approach that could assist the analyst in determining what the problem situation was and how to resolve it would be a significant step forwards in the world of information systems thinking.

Any approach that incorporates the concept detailed above requires the ability to fully embrace the concepts of self-adaptation and autonomy to ensure that it is able to accommodate any changes that may occur within the problem situation or operational environment as a whole. These new approaches have been incorporated into the development of a new generic framework that was designed to assist the analyst in determining what the problem situation was and how to resolve it.

### **1.3 Challenges Facing the Development of the Generic Framework**

Whilst IT systems have seen successful steps forward in developing more complex solutions it still requires the ability to evolve and to be able to deal with unexpected failures or conflicts. The same is true for the development of a generic framework, which also needs to fully incorporate the cultural, political and philosophical aspects that exist within organisations. This is derived from an appreciation that requirements for IT systems change considerably throughout their lifetime, meaning that for a methodological approach to implement appropriate solutions, it has to include functionality to deal with change. This was a core goal for the generic framework and its ability to provide appropriate solutions to any given problem situation.

Introducing a generic framework that includes management and controlling functionality represents a significant challenge, which requires the development of robust structures. Access and reason with information emerging from a variety of monitoring functions and determining whether to react or not must be covered.

The ability to fully support predictable and conditional triggers to facilitate change management based on the detection and prioritisation of problem situations, whilst coordinating conflict resolution strategies is crucial to its success.

The functionality to support negotiations to resolve failures or conflicts that lead to the selection of appropriate resolutions along with the ability to configure and reconfigure information systems by dynamically enabling a management and control



function to implement resolution strategies for the information system to self-adapt itself is imperative.

Incorporating normative models that are utilised for storing management policies, enforcing information system governance and authorising or vetoing proposed resolution strategies have been incorporated.

Whilst the issues discussed above were seen as challenges to the creation of a generic framework it is considered crucial that they be overcome to ensure that the aims and objectives of this research were achieved. That said, these are not the only challenges that must be overcome. The aim for the generic framework was to develop functionality that will allow it to cope with other significant challenges that occur.

To ensure that the challenges highlighted above were overcome it had been established that the robust structures also needed to be able to operate in a flexible manner. This approach will also encourage the adoption of self-adaptive and autonomic concepts.

To accommodate the self-adaptive and autonomic capacities of the generic framework must include three key functionalities, which were as follows:

- The management functionality contains control processes such as monitoring, diagnosis capabilities, repair functionality and adaptation strategies.
- The controller functionality includes information system control processes such as monitoring, repair strategies and reconfiguration capabilities.
- The control rules functionality can be accessed by both the management and controller and provides important details and definitions for boundaries, control and resolution strategies and repair plans.

The self-adaptive capacity of the generic framework required constant communication with the information system through effective feedforward and feedback loops, which assist it with its management and controlling duties.

## **1.4 Contributions Made to the Field of Information Systems Thinking by this Research**

The core contribution of this study has seen the development of a new generic framework that can be applied to any problem situation. This is due to its ability to cope with changing environments through its use of self-adaptive and autonomic concepts. The generic framework facilitates the ability to self-manage through containing functionality that can monitor, repair and reconfigure should any conflicts, failures or inconsistencies be highlighted in the environment within which it exists.

The creation of a new generic framework combines the philosophical approaches of two well established methodological approaches from the information systems thinking world. Whilst containing new functionality it should be noted that the approach adopted allows for both hard and soft information systems thinking to be incorporated into one methodology. This is a new development in the field of information systems thinking and provides the analyst with the opportunity to gain a full understanding and appreciation of problem situations from both perspectives.

The development of the generic framework has provided a new methodological approach within which the functionality of self-adaptation, autonomy and the capabilities from Soft Systems Methodology (SSM) and Viable Systems Model (VSM) can be embraced.

The generic framework is tested in a real world problem situation, which operates as a case study. Throughout the case study discussion references are made to publications made by the author at various conferences where the work carried out, and subsequent results, are demonstrated and assessed [5, 6, 7, 8, 9].

The generic framework incorporates the concepts of self-adaptation and autonomy. The generic framework is based on significant research into control, coordination, autonomic computing, deliberative systems and normative systems. In particular this work defines the requirements for a generic framework that is capable of governing itself and supporting self-adaptation through monitoring, failure or



conflict detection, diagnosing problem situations and implementing resolution strategies.

The management functionality concerns itself with managing failures or conflicts that have been detected. The management functionality itself contains the following capabilities:

- The monitoring functionality utilises a set of control rules to monitor behaviour and detects failures or conflicts.
- The diagnosis functionality involves the execution of the information system's control rules, which are activated by the detection of failures or conflicts, to identify the conflict type and provide a basis for the selection of resolution strategies.
- The repair model is specified preconditions and looks to provide operations that resolve a detected failure or conflict. These operations include notification, repair operators and appropriate resolution proposals. Whichever is chosen to resolve the conflict the success of the conflict resolution is stored within the management functionality.

The following subsections of this chapter will be discussed in this thesis and form the backbone of the generic framework that is developed and tested.

### **1.4.1 Conflict Management**

A monitoring and conflict management model must have the ability to collect and store the information that is required to support and guide the resolution strategies within the control process. The management functionality stores its system state and this is received later by the system monitoring model to check the system's states and starts the control process sequences should a conflict be found.

### **1.4.2 Control**

The controller functionality is responsible for establishing and managing the coordination of the overall functions and ensures that the communication channels are maintained and coordinated. The controller regularly checks the system state, stored previously by the management functionality. The system controller applies the appropriate strategy according to the difference between the current and desired state. The controller functionality contains three main models as discussed in this thesis.

### **1.4.3 Reconfiguration**

A system reconfiguration model that applies the required reconfiguration attached to the resolution strategy is a prerequisite. This is dynamically interpreted from establishing the requirements for the change and whether anything similar has been experienced before. If the problem situation has been experienced previously the success of the chosen resolution strategy at the time will be assessed. A decision will then be taken to decide whether to implement the previous resolution strategy or develop a new one.

### **1.4.4 Repair and Adaptation Strategies**

A system repair strategy model that determines when, where, and how the repair or adaptation is required. The repair strategies must consider the functions of the relevant information system, the operating environment and its attributes and properties. The generic framework's resolution strategies are used to evaluate the effect of various alternative solutions based on the Beliefs, Desires and Intentions (BDI) Model of deliberative systems, which provides functionality for self-adaptation allowing the generic framework to cope with changes that occur. This is introduced in Chapter 4 of this thesis.

The development of the contributions detailed above will be discussed in the following chapters of this thesis. The contributions made are discussed and evaluated in Chapters 10 and 11 of this thesis.



## **1.5 Thesis Structure and Organisation**

The thesis has been set out in eleven chapters so as to ensure due consideration can be given to the research conducted, demonstrate how the generic framework has been developed and how it has been tested.

Chapter 1 establishes the motivations for the research conducted, the challenges that were overcome and the new contributions to the field of information systems thinking.

Chapter 2 examines the main methodological approaches available in the world of information systems thinking. Appropriate methodologies for the creation of the generic framework are chosen to ensure that the generic framework contains pertinent functionality at its core and allows it to be applied to any given problem situation.

Chapter 3 provides a background to self-adaptive and autonomic approaches adopted in modern IT systems. This chapter aims at highlighting the potential for transferring these approaches over to the information systems thinking world.

Chapter 4 provides an in depth literature review of the philosophies and concepts that exist and the generic functionality that is inherent within them. This chapter also expands on the approaches discussed in Chapter 2.

Chapter 5 establishes the requirements for the generic framework that was to be created. The generic framework was developed so that it can be applied to different problem situations and demonstrates its generic capabilities. This chapter assesses the functionality that must be incorporated into the generic framework including static and dynamic management approaches to achieve the goal of true generic capacity.

Chapter 6 discusses the design of the generic framework and establishes how the two chosen methodologies will link together. This chapter demonstrates the complementary functionality between the two chosen methodologies and their

combined ability to support the goal of self-adaptation and autonomy, which could not be provided through other methodological approaches.

Chapter 7 details the development of the generic framework. The aim of this chapter is to highlight how the generic framework was constructed to ensure the 'whats' and the 'hows' of problem situations could be understood and resolved.

Chapter 8 establishes the right environment for the generic framework and provides details of its implementation in a case study. This chapter also establishes a structured definition of the problem situation within a large organisation and provides the baseline within which the functionality of the generic framework would be able to facilitate the need for change.

Chapter 9 tests the generic framework using a large organisation as a case study, which has been chosen as it required a new information system to be implemented to resolve a long standing information starvation problem.

Chapter 10 presents an evaluation of the experiments using the generic framework. The evaluation discusses the findings gathered from experiments along with experience gained and provides a proof of concept.

Chapter 11 provides conclusions and a summary along with details of future work.



## **2. Methodologies Employed**

### **2.1 Introduction to the Methodologies Employed**

This chapter establishes the goal of identifying appropriate methodologies to utilise within the generic framework. In order to develop a generic framework it is first important to understand the process that is required to undertake such a task. There are two main functions of a self-adaptive system, which must be taken into consideration when developing the framework, which are; the capability to determine '*what*' changes are taking place and '*how*' to take the necessary corrective action. An in depth literature review has been conducted so as to ensure that knowledge of the field is extensive and comprehensive. It is vital to the development of a generic framework that self-adaptive systems concepts are reviewed so as to take into account current and emerging control structures.

Having gained an understanding of the processes involved it is logical that the next steps to take are the design, implementation and experimentation of the generic framework. To begin with, it must be understood and appreciated that there are numerous theoretical approaches for self-adaptive information systems as evaluated by Andriole [10] and Murray-Smith [11].

Peter Checkland's Soft Systems Methodology [12] will be utilised to underpin the concepts incorporated into the generic framework. Whilst the soft approach adopted by SSM will allow the framework to determine '*what*' changes are happening to the new information system it will not be able to decide '*how*' to take corrective action if necessary. Because of this, a hard methodology will be employed to control the decision making process that is fundamental to any information system.

In the world of public sector information system development and implementation Prince2 [13] is the Government standard approach. An investigation has been undertaken to determine which hard approach is most suited to the framework being created. Other approaches such as Effective Technical and Human Implementation of Computer-based Systems (ETHICS) and Stafford Beers' Viable

Systems Model, by Espejo [14], have been evaluated. It is essential that the hard approach adopted for the development of the framework must be able to link in with SSM to ensure that the generic framework can adapt to changing environments in instances where an appropriate decision must be made.

This chapter introduces the methodologies evaluated and discusses their concepts, philosophies and beliefs. Once the methodologies have been selected for integrating, and forming the basis of the generic framework, a discussion will be presented detailing how they can be linked together and how their concepts, philosophies and beliefs complement each other.

## **2.2 Soft Systems Methodology**

SSM, as developed by Checkland [15], is an approach to solving complex unstructured human problem situations based on holistic analysis and systems thinking. SSM is a participatory methodology that helps different stakeholders to understand each other's perspectives. It focuses on creating the human activity systems and human relationships needed for an organisation or group to achieve a common purpose.

The SSM premise derived by Checkland [16] is that if people participate in the process of finding out about the problem situation and learning about ways to improve it, then they are more likely to understand the improvements being suggested, feel ownership of them and be committed to change. This was a major factor SSM being chosen.

SSM is based on a seven-stage process, called the Lancaster Model (with full details of each stage detailed in the generic framework's user guidance in the Appendix), which moves from clarifying an unstructured or messy problem situation through designing ideal or conceptual human activity systems that would help improve the situation. These conceptual models are then compared with the problem situation in order to identify desirable and feasible change. The methodology



integrates thinking about the logic of how to improve a situation with what is socially and politically feasible.

A Root Definition in SSM is a concise, tightly constructed description of a Human Activity System which states what the system is; what it does is then elaborated in a conceptual model which is built on the basis of the definition. Every element in the definition must be reflected in the model derived from it. A well formulated root definition will make explicit each of the elements in the mnemonic **CATWOE** as detailed below:

C (Customers)... who are the direct victims or beneficiaries of the transformation?

A (Actors)... who would do these activities?

T (Transformation process)... what input is transformed to what output?

W (World)... selected world view; a system for?

O (Owner)... who could abolish this system?

E (Environmental constraints)... what does this system take as given?

Given that SSM has complete flexibility when developing an understanding of problem situations, along with providing a structured architecture to assist the analyst, it has been deemed suitable to utilise this methodology in the generic framework. The ability to develop a strong understanding of complex problems including human interaction and cultural issues through adopting a holistic view is key to self-adaptive and autonomic functionality.

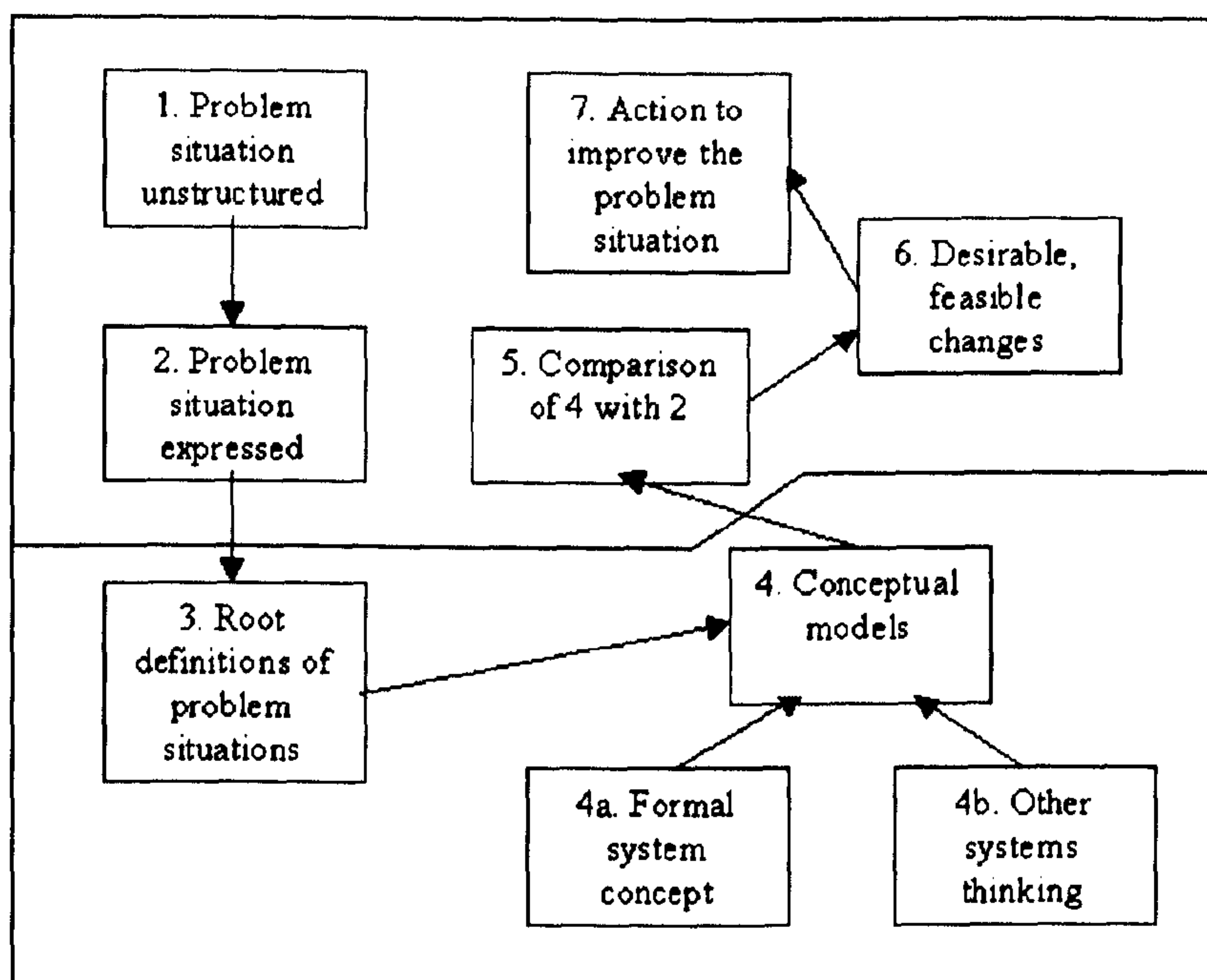


Figure 1: Adaptation of the Lancaster Model produced by Checkland [17].

## 2.3 Viable Systems Model

The Viable Systems Model (VSM), developed by Beer [18], is a model of the organisational structure of any viable system. A viable system is any system which is organised in such a way as to meet the demands of surviving in changing environments. One of the prime features of systems that survive is that they are adaptable. The VSM expresses a model for a viable system, which is an abstracted cybernetic description that is applicable to any organisation that is a viable system. The first thing to note about the cybernetic theory of organisations encapsulated in the VSM is that viable systems are recursive; viable systems contain viable systems that can be modelled using an identical cybernetic description as the higher (and lower) level systems in the containment hierarchy. Stafford Beer expresses this property of viable systems as cybernetic isomorphism.



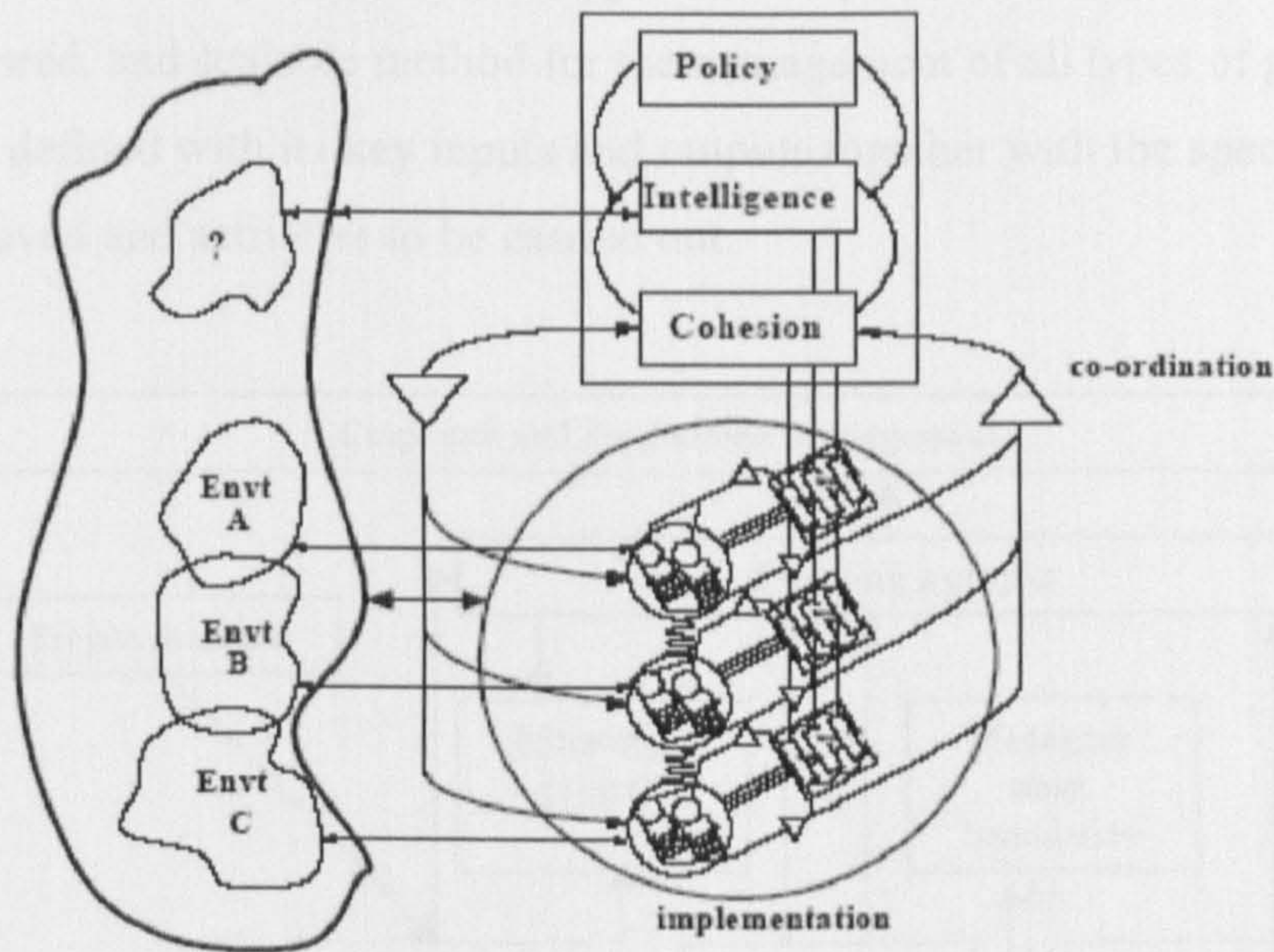


Figure 2: Adaptation of the Viable System Model [19].

The diagram above is the VSM in its entirety. Although the diagram may look complicated to people who have not have seen it before it is essentially fairly simple. The bottom right part of the diagram (implementation) shows the viable systems going about their work in a semi autonomous fashion. The left hand side of the diagram is the environment, which is constantly being monitored by the viable systems. Should any significant change appear in the environment the ‘Cohesion’, ‘Intelligence’ and ‘Policy’ parts of the VSM will make a decision on any alterations which need to be made, which are then implemented.

## 2.4 Prince2

Prince2, implemented in public sector organisations by the UK Government [20], is a project management methodology designed to cover the range of disciplines and activities required within a project. It is a structured method for effective project management. Prince2 sets out a definite organisation structure for the project management team and is accompanied by detailed descriptions of responsibilities and tasks for each role in the Prince2 organisation structure. Emphasis is also placed on the feasibility of the project to be undertaken and the identification of all the project interests before the project is undertaken.



Prince2 is a process based approach to project management providing an easily tailored, and scalable method for the management of all types of projects. Each process is defined with its key inputs and outputs together with the specific objectives to be achieved and activities to be carried out.

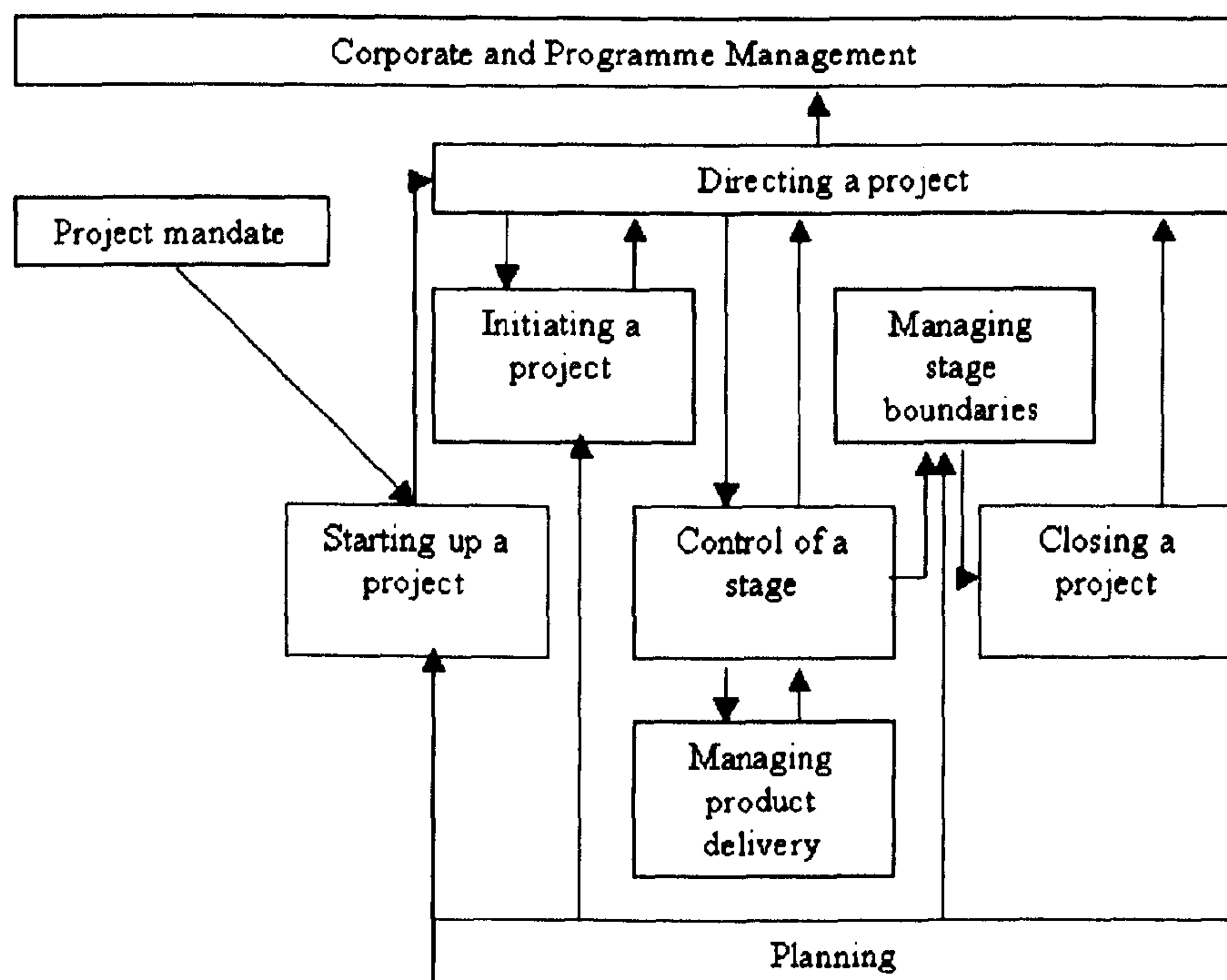


Figure 3: Adaptation of the Prince2 process model [21].

Having assessed the philosophy that underpins Prince2 it was decided not to incorporate it into the development of the generic framework as it was not deemed suitable for the generic framework in this research. The methodology itself is geared towards the successful completion of well structured and defined projects. In fact, Prince2 is seen as the Government standard methodology for projects run within public sector organisations. However, it is felt that Prince2 would not be able to link in with SSM and would not be fully functional when operating in a constantly changing environment which is potentially poorly defined and requires a certain level of autonomy.

## **2.5 Effective Technical and Human Implementation of Computer-based Systems (ETHICS)**

ETHICS, devised by Enid Mumford at Manchester University in 1979, is a methodology based on the participative approach to systems development. In addition, it encompasses the socio-technical view that for a system to be effective the technology must fit closely with the social and organisational factors. In particular, this means that an improved quality of working life and enhanced job satisfaction of the users must be a major objective of the systems design process. This is not simply to guard the interests of the users in the introduction of computing and technology, although this is obviously of major importance, but it is an essential prerequisite to achieve effective systems as far as the organisation and its management is concerned. To support her case, Mumford [22] points to the failure of many traditionally-performed system implementations, where technical and economic objectives were the only consideration.

The philosophy of ETHICS is different from most information systems development methodologies and is also explicitly stated, which is not common in most methodologies. The philosophy is one which has evolved from organisational behaviour and perceives the development of computer systems not as a technical issue but as an organisational issue which is fundamentally concerned with the process of change.

ETHICS has not been chosen for this investigation as although it does encompass social and organisational factors it is mainly aimed at the successful implementation of computing and technology related solutions to problem situations. At its core ETHICS concerns itself with implementing an IT solution to problem situations, which is not always appropriate when developing new information systems. This is imperative for the generic framework and therefore it is possible that conflicts could be experienced should ETHICS be chosen for the generic framework. Whilst this methodology's effectiveness is largely underrated it is felt that it would not be successfully linked in with SSM as there would be some potential clashes with the philosophies utilised by each methodology.



## 2.6 Multiview Framework

Multiview was defined in 1985 and has been refined since to become an influential approach to systems development. As evaluated by Laws [23] it has soft and hard aspects and, as a contingency approach, is not prescriptive but adapted to the particular situation in the organisation and the application. Observations and reflections on Multiview in action over the last ten years together with more recent literature based on, for example, holism, emergence, multi-causality, ethical analysis and technology foresight, form the basis for a new definition of the Multiview framework. Changes in the domain of information systems are also taken into account. Away from centralised technology, long lead times and hierarchical organisations, towards networks, new organisational forms, business processes, informational products and services, and the removal of time and space constraints on human activity.

The framework is shown below in its entirety:

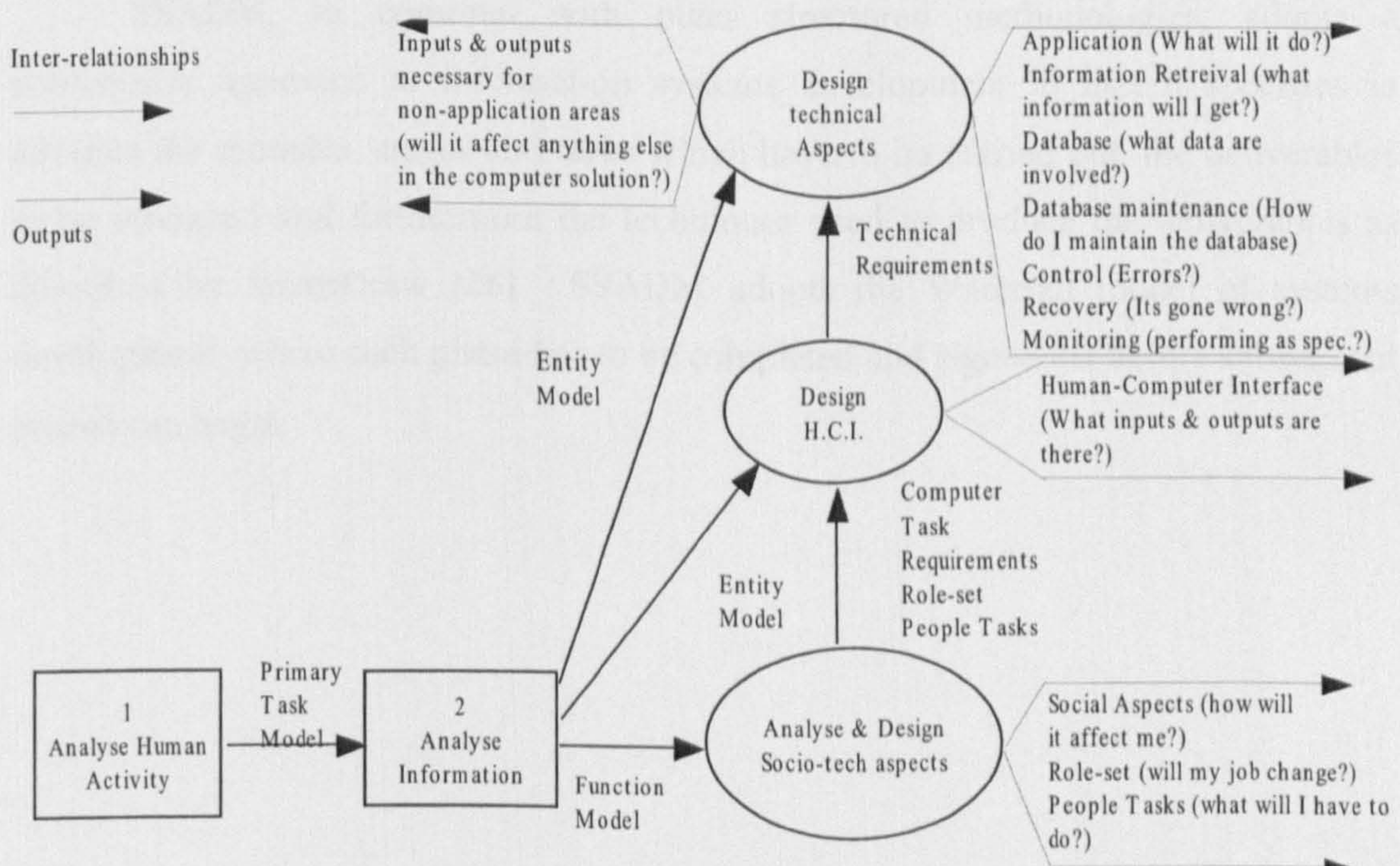


Figure 4: Adaptation of the Multiview framework [24].

As is the case with ETHICS this methodology has not been chosen as although it does encompass social and organisational factors it is mainly aimed at the



successful implementation of computing and technology related solutions to problem situations. In fact, this methodology does include both a hard and a soft approach to developing solutions to problem situations. However, it is felt that it would not be successfully linked in with SSM as there would be some potential clashes with the philosophies utilised by each methodology and that Multiview would not be able to fully embrace the concepts put forward by autonomic and self-adaptive systems thinking. In recent year Multiview 2 has been developed to attempt to provide a three way relationship between analysis, the problem situation and the methodology itself [25]. However, the potential clashes between the philosophies with SSM and Multiview 2 are still in existence. At its core is functionality for developing human computer interaction. Whilst this can develop suitable IT solutions this is not always appropriate when developing new information systems. This is imperative for the generic framework and therefore it is possible that conflicts could be experienced should Multiview be chosen for the generic framework.

## **2.7 Structured Systems Analysis and Design Methodology (SSADM)**

SSADM, in common with other structured methodologies, adopts a prescriptive approach to information systems development in that it specifies in advance the modules, stages and tasks which have to be carried out, the deliverables to be produced and furthermore the techniques used to produce the deliverables as described by SmartDraw [26]. SSADM adopts the Waterfall model of systems development, where each phase has to be completed and signed off before subsequent phases can begin.

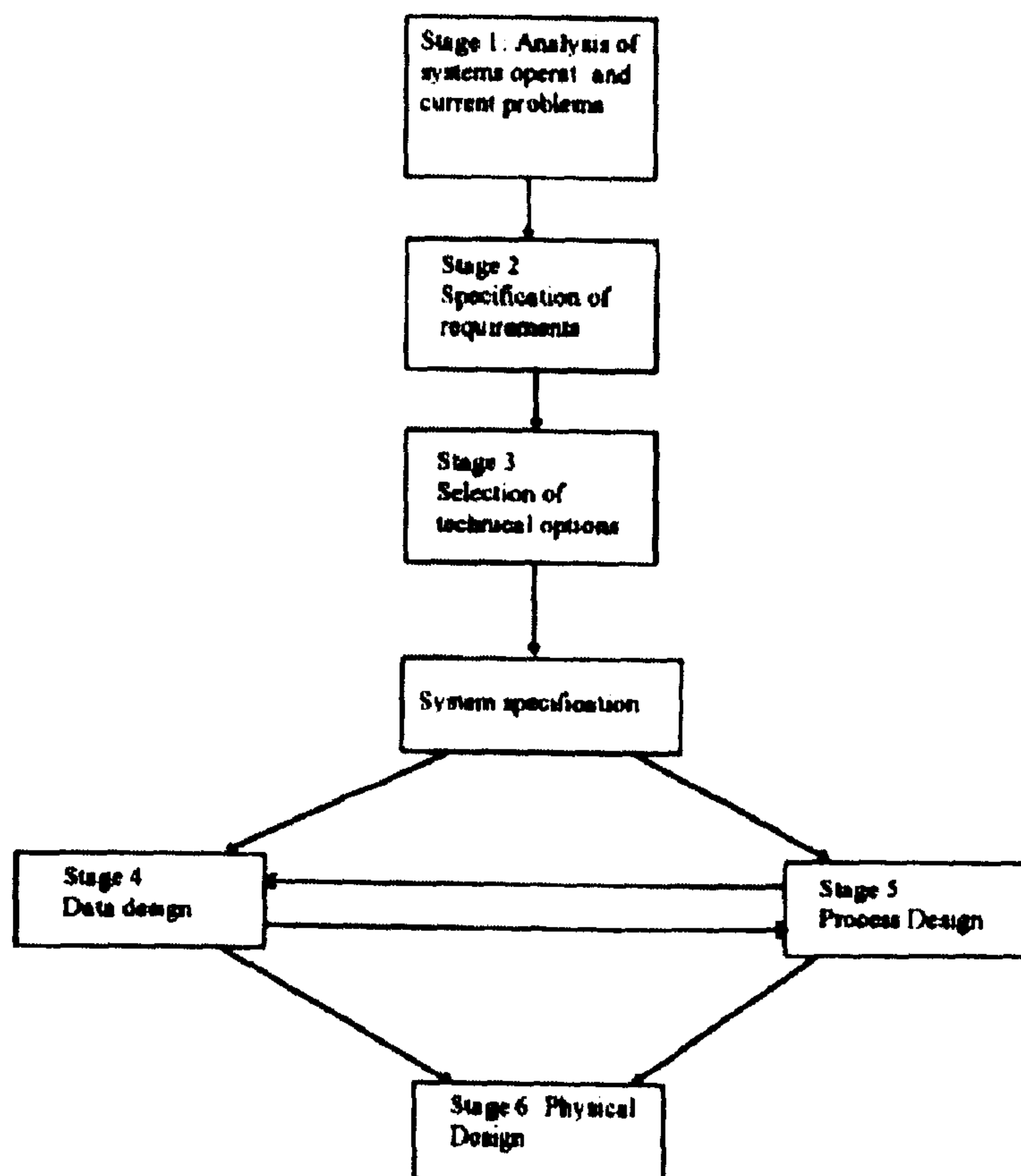


Figure 5: Adaptation of the SSADM framework [27].

SSADM consists of several main stages (which are broken down in several sub-stages). The Feasibility Study involves a high level analysis of a business area to determine whether it is feasible to develop a particular system. Data Flow Modelling and (high-level) Logical Data Modelling can be used as technique during this stage.

The Requirements Analysis stage requirements are identified and the current business environment is modelled, business system options are produced and presented. One of these options will be chosen and refined. Data Flow Modelling and Logical Data Modelling can be used as technique during this stage.

In the Requirements Specification the functional and non-functional requirements are specified as a result of the previous stage. Data Flow Modelling, Logical Data Modelling and Entity Event Modelling can be used as techniques during this stage.

In the Logical System Specification the development and implementation environment are specified, and the logical design of update and enquiry processing and system dialogues are carried out.

In the Physical Design the logical system specification and technical specification are used to create a physical design and a set program specifications.

Due to the inherent structured approach adopted by SSADM the decision has been taken not to incorporate it into the development of the generic framework. The methodology performs very well when utilised on its own, however, linking it to SSM would not work in this case.

## **2.8 Hard and Soft Methodologies**

As has been covered in the previous sections of this chapter it can be seen that the world of information systems thinking contains a diverse range of methodologies and approaches. Essentially the methodologies can be grouped into two definitions due to the philosophies that underpin them.

Hard methodologies are defined as being methodologies that *"start from a requirement and produce a product"* by Veryard [28] and, as such, only consider what has been stated in the requirements. In most cases this does not incorporate cultural, philosophical or political beliefs of people involved and only looks to establish a resolution to the problem situation being experienced.

Soft methodologies are defined as being methodologies that *"start from an open situation, where specific problems and opportunities have not yet been identified or agreed, and produce a plan for improving the situation."* as discussed by Lyytinen [29]. Although an agreed plan for improving the problem situation can be identified it does not define how to implement a resolution strategy.

In essence, the two different approaches that methodologies incorporate establish the 'whats' and 'hows' of problem situations, however, there is not one approach that combines the two. This is why an evaluation will be conducted in the



following sections of this chapter to establish an approach that will incorporate the ‘whats’ and the ‘hows’ of the problem situations, which will form the basis of the generic framework.

## **2.9 Choosing the Methodologies**

In this chapter it has been discussed that SSM will be chosen as the soft methodology as its philosophy complements the requirements of the generic framework perfectly. The requirements discussed in Chapter 5 discuss the necessity for there to be flexibility and the ability to adopt other systems thinking, which SSM is more than capable of doing. The majority of methodologies and the philosophies that underpin them have been evaluated. The methodology descriptions also discuss whether they would work with SSM and, as a consequence, whether they will be chosen for the generic framework.

The outcome of the evaluation of the methodologies shows that VSM is the most appropriate for integrating with SSM and the development of the generic framework. Both methodologies share similar philosophies whilst still ensuring that they contain the functionality required of them to meet the demands of hard and soft methodologies.

## **2.10 Linking SSM and VSM**

Information systems thinking assumes that when studying a real world problem situation there are some characteristics that cannot be identified by studying its parts separately, but are only visible when combining the parts. This is what Checkland [30] refers to as “*emergent properties*”. Checkland relates the idea of emergence to the concepts of complexity and hierarchy. It is argued by Mirijamdotter [31] that organised complexity is the subject matter of the discipline systems, and organisation of the complexity can be made in terms of hierarchical levels. As Checkland states:

*“The general model of organised complexity is that there exists a hierarchy of levels of organisation, each more complex than the one below, a level being characterised by emergent properties which do not exist at the lower level.”*

A second suggested foundation, closely related to different hierarchical levels of complexity is the concepts of communication and control. These concepts are borrowed from the field of cybernetics by Ashby [32]. For Beer [33], cybernetics is the *“science of effective organisation.”* In his work on the Viable Systems Model, Beer identifies essential functions and associated information processes, which are required for a system to function effectively and be viable. Viability means being capable of living and of maintaining a separate existence – in information systems terms this is being able to survive by self-regulation in a changing environment according to Mirjamdotter [34].

In the finding out phase the emphasis in SSM is on getting as ‘rich a picture’ as possible of the problem situation. The guidance that is proposed by von Bulow [35] is to look for structure and processes within the climate of the problem situation.

The focus of SSM’s second phase, in which models of relevant systems are built, is on information systems thinking. The metaphor of an adaptive whole, with a hierarchical structure, emergent properties on each hierarchical level, and processes of communication and control, is used to structure the thinking as discussed by Avison [36].

Since the SSM modelling phase is based on systems thinking, as is the philosophy of VSM, there are many similarities and relations found in this phase. There are three main similarities between the two methodologies. The concept of an adaptive whole used in SSM is also a cybernetic concept on which VSM is based. The emphasis is on adequate information for the processes of communication and control. This is significant by the inclusion of feedback flows, necessary to guarantee the stability of the system and its ability to be viable in a changing environment.

A core idea of SSM’s systems thinking is the concept of emergence and hierarchy. This concept is also found in the recursive structure of cybernetics.



The concepts of CATWOE, root definition and conceptual models relate to the elements of the VSM. Mingers [37] states that the concepts of CATWOE, root definition and conceptual models imply an input to output transformation.

It can be seen that there are significant overlapping philosophies for the two chosen methodologies to work together and be fully integrated into the development of the generic framework.

Although the two methodologies have overlapping philosophies they also have the flexibility to adapt to change. In the case of SSM the Lancaster Model contains the ability to accommodate other system's thinking. The point at which this capability is present is at the Conceptual Model step of the Lancaster Model. In effect this step of the Lancaster Model is the first step towards the transition between the information systems thinking and the real world – or in other words the 'whats' to the 'hows' of the problem situation. VSM has the capacity to deal with change through its control and management functionalities and its ability to understand both the internal and external environments within in which it exists. Although the two methodologies utilise different approaches, as would be expected with one being hard and one being soft, it can also be seen that their respective functionalities would complement each other in a generic framework.

## **2.11 Chapter Summary**

The aim of this chapter has been to provide a review of the different existing methodologies available in the world of information systems thinking and the differences that can be found between hard and soft approaches.

This chapter details why SSM and VSM have been chosen as the two methodologies to incorporate into the generic framework. The methodologies that were evaluated all had potential benefits to offer the generic framework, however, SSM and VSM complement each other to such an extent that some of their philosophies, beliefs and even functionalities overlap with each other.

The selection of appropriate methodologies will ensure that the generic framework is capable of reacting positively and effectively to change. It is also essential to the success of the generic framework that the two chosen methodologies had the capacity to fully embrace the concepts of self-adaptation and autonomy. Through linking SSM and VSM, with their respective control and management functionalities ensures that self-adaptation and autonomy are incorporated into the generic framework.

The following chapter discusses the concepts of self-adaptation and autonomy. This capability is crucial to the linking of the two methodologies to lead to the creation of the generic framework, which in turn will develop new information systems to resolve long standing problem situations.

## **3. Background**

### **3.1 Introduction to the Background**

In order to gain an understanding of self-adaptation and autonomic concepts for inclusion in the field of information systems thinking it is necessary to gain an understanding of such concepts as computing principles. These approaches are currently utilised in the world of computing and software development although the concepts and philosophies that underpin them can be translated into information systems thinking. This chapter outlines the current and emerging approaches and techniques that should be considered important to the understanding and development of the generic framework. IBM has invested considerable time and money in this field and the findings from the research conducted is included in this chapter.

This chapter provides an account of self-adaptation and autonomy in computing. Given the number of research projects and strands have taken place over recent years the terminology is different depending on which papers are being reviewed. This chapter presents one particular term for each item of functionality reviewed in order to provide consistency.

### **3.2 Self-Adaptive Software**

It is not possible to account for every potential change in an environment. As a consequence, any computing system that is designed to cope with such changes must be tolerant and adaptive. Ad hoc approaches are adopted today but the severity of faults increases as systems become more complex. In order to combat this it is proposed in this thesis to attempt to find a reasonable methodological approach to find a solution to this problem. Badr [38] discusses that systems must have the capacity to learn and evolve in order to deal with changes in an environment.



One definition of self-adaptive software was provided in a DARPA Broad Agency Announcement on Self-adaptive Software [39] is as follows:

*“Self-adaptive software evaluates its own behaviour and changes behaviour when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible”.*

The implication in the above statement is that the software system has multiple ways of achieving its goal. The system requires sufficient knowledge of its own capabilities in order for it to make effective changes during the runtime process. Systems such as this require functionality that enables it to evaluate its own behaviour and performance in its environment. In addition to this the ability to reconfigure and ‘re-plan’ activities in order to improve its own operation is imperative. Self-adaptive software would benefit from being granted a set of tools that are capable of working with the major system functions. A description of the core functions should be incorporated so that scheduling and prioritising of those major functions can be utilised when responding to changes in the environment. The ability to measure inputs and outputs of major functions and also generate suitable resolutions is fundamental.

The ambition of self-adaptive software is to delegate the responsibility and maintenance to the software itself. This is highlighted by A. Laws *et al.* [40] as shown below:

*“...to devolve some of the responsibility for evolutionary activity to the software itself. Essentially, this requires embedding equivalent elements of the human software evolution process in the software itself, thus allowing autonomous adaptation to local conditions during runtime. Effectively, such software must be capable of detecting the need for change, either to address changing external conditions or for internal performance-related reasons, determine which elements to change and how they should be changed, planning and enacting the change and finally verifying the effectiveness and robustness of the resulting solution.”*

Research carried out over the past few years suggests that there are three main directions that relate to self-adaptive software. The three directions are:

- Control systems theory;
- Dynamic planning systems; and
- Self-aware systems.

The three research directions are outlined in the following sections of the chapter.

### **3.2.1 Control Systems Theory**

IBM defines control systems theory as being the mathematical analysis of the systems and mechanisms for achieving a desired state under changing internal and external conditions [41]. This approach is reinforced in the modern control theory of mathematics [42]. Self-adaptive software for systems in the engineering environment is incorporated into this research direction.

The main three elements considered by control theory are:

- The external environment;
- The productive element of the system that is concerned with providing products or services for that environment; and
- A model-based self-control unit that ensures the system meets the policies and norms of the environment [43].

IBM has concentrated most of its research into the flexibility of each system's control element. At present, IBM is particularly working towards the development of a hierarchy of complex control models. These control models are designed to assist the system by increasing its adaptive capabilities.



### **3.2.2 Dynamic Planning Systems**

Despite research being carried out on mainly engineering systems it is probable that all forms of system will be included in the future. All sizes of systems will be affected by this research and autonomic computing will have implications for them all. IBM predicts that small scale systems, which contain elements of autonomic computing, will be relatively simple. The intention for such systems would be for the autonomic elements to perform similar activities with similar functions over fairly lengthy periods. It is felt that at higher levels the autonomic computing elements will operate in a more dynamic environment whereby only its original policies and goals will remain constant. Here the autonomic elements will constantly alter the methods utilised to interact with system functions. The possibility of having to change its methods and approaches almost every second is quite likely in an ever changing environment.

There are likely to be two main common architectures for an autonomic element. A Functional Unit performs basic functions that the element provides such as storage, Web services and database functions. The Management Unit will oversee the operation carried out by the Functional Unit and ensure that it has all necessary resources to perform the required tasks. This unit configures and reconfigures the system to adapt to changes in the environment. This unit is also responsible for negotiating with other autonomic elements within the system [44].

### **3.2.3 Self-Aware Systems**

To highlight new and emerging concepts job adverts, in particular for Java programmers ([www.monster.co.uk](http://www.monster.co.uk), August 2005), have started to use the words 'intelligent' and 'self-aware' when describing the types of systems they expect the applicant to have experience of working with. The use of such words differs from the way they are used in daily English conversation. The common feeling amongst computer programmers is that self-awareness relates to reflective software.



Reflective software allows the system to have the ability to monitor itself during runtime and execution [45]. Opinion is split with regards to what reflective software is. Some computer programmers feel that reflection is self-aware software. Some feel that reflective software is a term for a system that is able to create a representation of itself and its purposes and goals [46]. Different branches of Computer Science have interpreted the details of reflective software differently. For a comparative study, see [47].

In essence, self-aware systems are developing a new kind of software that improves itself by learning from experience gained. This enables it to be flexible and naturally adapt to changing circumstances.

In terms of being able to be aware of its own environment, and monitor whether any action is required by the self-adaptive functionality of the generic framework, this is seen as being an important capability. Inherent within the generic framework must be the ability to learn from experience and as a result this must be incorporated into the self-adaptive functionality.

### **3.2.4 Self-Adaptive Software Compared to Control Systems Theory and the Principles that Underpin the Approaches**

The development of self-adaptive software has been compared to a control system. This requires robust and flexible programming, which incorporates new techniques in order to ensure that a control system resides within its structure. Control systems theory puts forward such ideas for self-adaptive software as highlighted by P. Robertson *et al.* [48].

Self-evaluation, and interpreting the results, is the most important problem that must be overcome in self-adaptive systems. Clarke *et al.* [49] stated that they feel that self-adaptive software should always be monitoring the difference between the current state and its desired state. The desired state is determined by operational requirements and system goals. Once the differences between the current and desired states had been assessed notification for areas of potential improvement were provided. Clarke

*et al.* base their paper around the concepts relating to the transition from software maintained by humans, who had the responsibility for testing, evaluating and updating systems to automated processes. The paper develops their theories by suggesting that automated and continuous self-evaluation is essential for self-adaptive software.

A. Meng [50] in “On Evaluation Self-Adaptive Software” defines the relationship between control systems theory and self-adaptive software. Meng develops a model of self-adaptation based around control systems theory principles. The feedforward and feedback control paradigms from control systems theory are incorporated into this model. The feedforward and feedback control paradigms also appear in Beer’s Viable Systems Model. This considers that self-adaptive software consisting of the feedforward elements that provide the specification of the software and its predictability and the feedback component that receives feedback from its environment. Meng correctly points out that it is possible for such a model to be used in many different aspects with self-adaptive software. A model such as this one should be able to cope with the introduction of new software engineering principles or approaches.

Meng discusses the evaluation of self-adaptive software systems based on their different aspects and provides four key points [51]. These four key points are as follows:

- As a new programming paradigm, reflection programs could modify themselves and change their behaviour, and as such are close to the concepts of self-adaptation. However, such programs cannot always determine when and what the program needs to modify itself. Self-adaptive software generates evaluators to check the deviation between the current state and its goal state and adapts to maintain its stability and robustness.
- As a new architecture style, self-adaptive software needs to formalise the feedforward and feedback controller concepts and configuration, which are system structural components that are part of the Architecture Description Language (ADL) [52]. The ‘configuration’ and ‘controller’ architecture



description vocabulary in self-adaptive software maintains stability when the system transfers from one configuration to another.

- As a new modelling paradigm, reconfiguration in terms of self-adaptive software concepts uses adaptive control to allow the system to switch to a control regime based on the current situation. Self-adaptive software transfers the feedforward process from the model to the executable and synthesises, however, the feedback process transfers from execution to reconfiguration and hence re-synthesises.
- As a software engineering principle, a software system can be seen as a control system incorporating adaptation and reconfiguration functionality based on adaptive control theory, and generalises the control model as a concept of algorithm selections in software engineering. The system is provided with different algorithms and software adaptation becomes choosing a suitable algorithm for a particular environment.

The creation of the generic framework will look to incorporate the functionality outlined in this chapter into information systems thinking and therefore the methodologies chosen for this study must be capable of accommodating such functionality.

### **3.2.5 Self-Adaptive Research Directions**

Significant amounts of research have been conducted in the field of self-adaptive software. Transient management is one key area of research with self-adaptive systems. The aim of transient management is where self-adaptive software reconfigures itself to ensure the robustness and performance of the system. G. Simon *et al.* [53] define reconfiguration as “any modification or change in system parameters”. Although modifications may be robust and reliable there is sometimes a tendency for some undesirable transient effects to occur. G. Simon *et al.* [54] investigated transient management and discussed two important issues.

Firstly, management depends on the suitable selection of a structure that features the transient properties. A formal definition of reconfiguration transients is



“the difference between the measured value in the reconfigured system, and the idea value in the reference system. The reference system is a hypothetical system, which operates in the new model for a long time.” [55]

The second issue is establishing control functionality to support the transient’s management. This is a model based generative technology called ‘Model-Integrated Computing’ [56], and is applied to self-adaptive software.

One of the areas of research on self-adaptive software has progressed at the programming language level. For example, there are now exception handling algorithms supported by modern programming languages. Adaptive mechanisms built in to the language architecture can provide a solution to problems. In these mechanisms systems architecture models are maintained and used as a basis for system reconfiguration. The architecture provides a view of entire systems as an abstract model. Integrity constraints can be seen clearly via its architecture.

Research on self-adaptive software based on architecture is progressed by many groups including David Garlan and Richard N. Taylor [57]. The diagram below shows a mechanism of self-adaptive software based on its architecture.

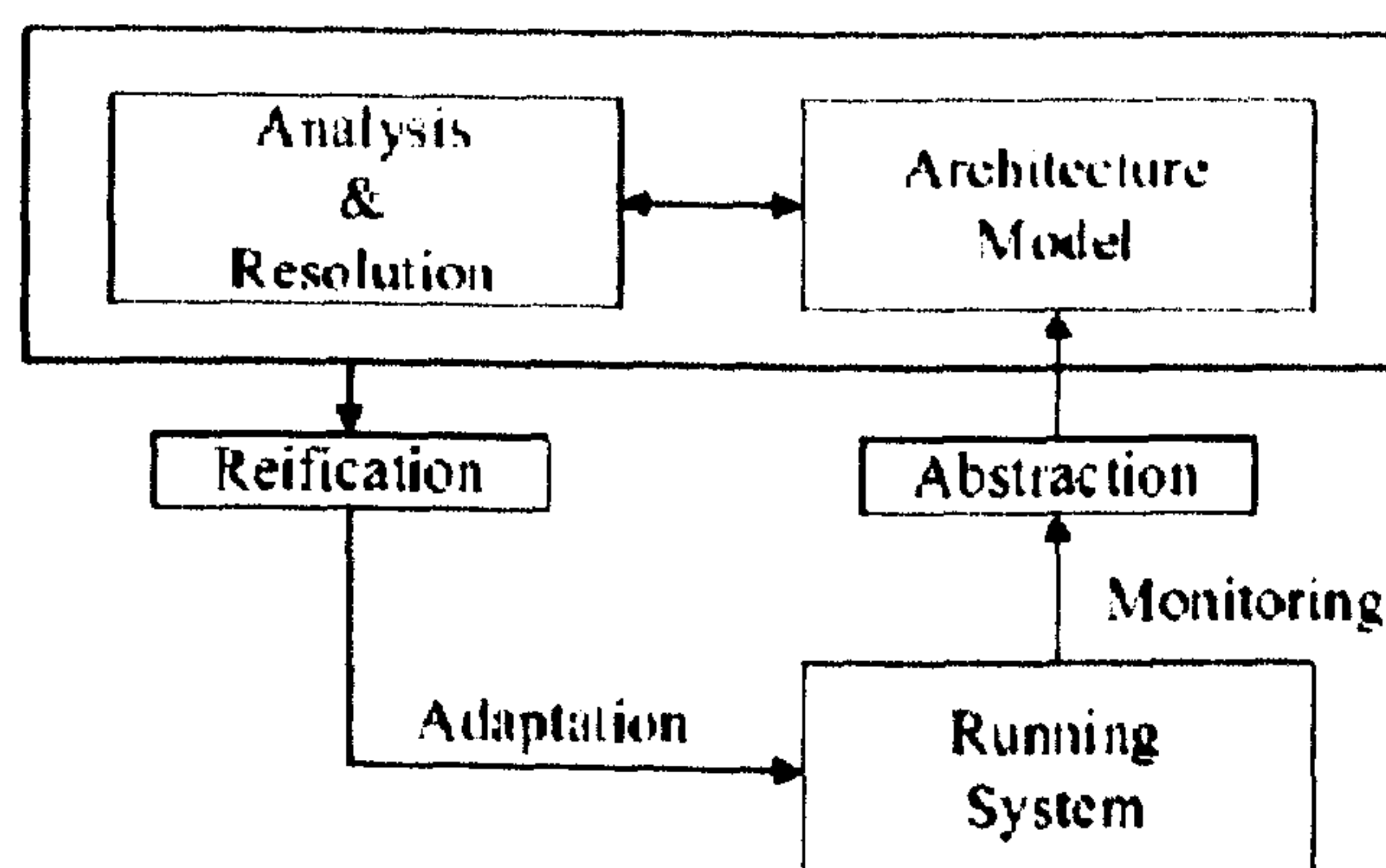


Figure 6: Mechanism of self-adaptive software according to Garlan and Taylor’s architecture [58].

### **3.2.6 Self-Adaptation and Reflective Architectures**

Following on from the concept of reconfiguration is reflective architecture. A reflective architecture enables programs to evaluate their own structural, behavioural and computational state to accommodate any changes in resources or the environment within which they operate if any failures or conflicts are detected. Reflective architectures provide the ability to write such programs but they do not describe how this is to be achieved or what functionality of the architecture should be used to achieve it [59]. Self-adaptive software has a computational model to compare the current program with the intended program and takes any corrective action it considers necessary to return it to the desired state. I. Shaul [60] described the use of reflective architectures incorporated within self-adaptive software concepts for the networking of computers. The network was able to cope with changes in its operating environment and further development of the concept is ongoing.

### **3.3 Autonomic Computing**

The concept of autonomic computing is an approach of self-managing computing systems that requires the minimum amount of human interference [61]. The philosophy that underpins autonomic computing will be incorporated into the development of the generic framework as it complements the functionality of self-adaptation.

Autonomic computing controls the key functionality of a system and is able to coordinate and manage itself while hiding the complexity from the end users. Autonomic computing should always have the ability to allow users to override any decisions it may take however. Autonomic computing systems have the ability to manage themselves and they have the capacity to dynamically adapt to changes their operating environments within defined boundaries. Systems with self-managing capabilities can perform operations or take corrective action in response to changes in their operating environments. According to IBM the main characteristics of an autonomic system include the concept that an autonomic computing system needs to 'know itself' and its components must also possess a system identity. It also requires detailed knowledge of its components, current status, ultimate capacity and all



connections to other systems, to govern itself. This extends to the theory that an autonomic computing system must configure and reconfigure itself under varying, and even unpredictable, conditions.

According to IBM an autonomic computing system never settles for the status, it always looks for ways to optimise its workings. It maintains its constituent parts and fine-tunes workflow to achieve predetermined system goals. An autonomic computing system must be able to recover from routine and unusual events that might cause some of its parts to malfunction, and discover problems, then find an alternative way of using resources or reconfiguring the system to keep functioning smoothly.

An autonomic computing system must be expert in self-protection. Also it must detect, identify and protect itself against various types of attacks to maintain overall system security and integrity.

Autonomic computing systems must know its surrounding environment and act accordingly. It will find and generate rules for how best to interact with neighbouring systems. It should negotiate the use by other systems of its utilised elements, changing both itself and its environment in the process of adapting. While an autonomic computing system is independent in its ability to manage itself, it must function in a heterogeneous world as well.

An autonomic computing system anticipates the optional resources needed while keeping its complexity hidden without involving the user in that implementation [62].

In the field of autonomic computing both e-sourcing and GRID computing [63] have started to embrace the functionality on offer. It is felt that in the next few years the concepts of autonomic computing will be embraced in a much larger scale than it currently is. However, the philosophies that underpin autonomic computing can be incorporated into the generic framework that will be developed for this study and will become a useful introduction to the world of information systems thinking.

### 3.3.1 The Autonomic Computing Architecture

Taking into account the points highlighted in the previous section of this chapter it can be deduced that there is a standard set of functions and interactions which govern the management of the computing system and its resources. This is the autonomic computing architecture and is represented by a control loop, shown below, that acts as a manager of the resources through monitoring, analysis and taking necessary action based on a set of policies and rules.

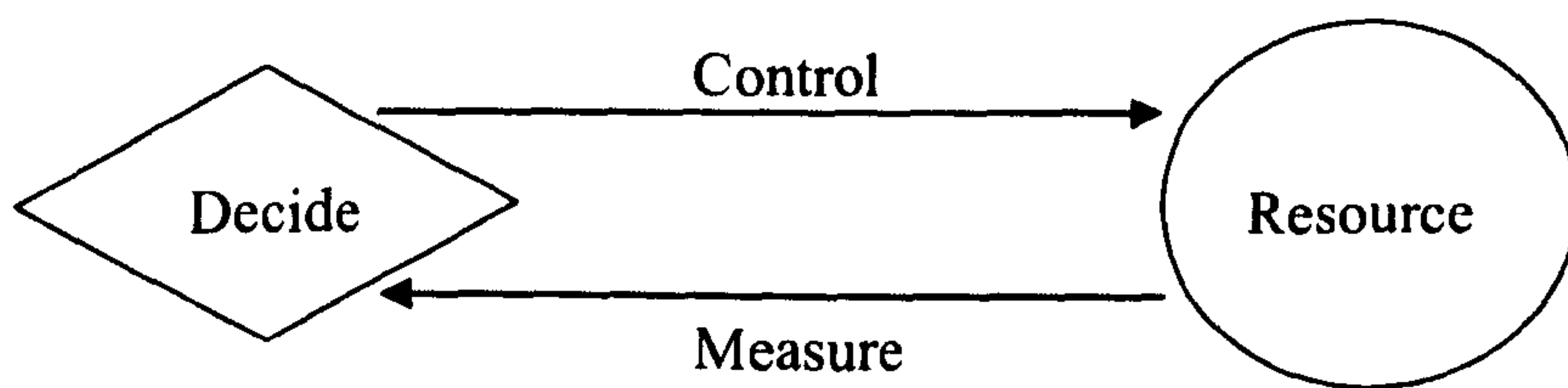


Figure 7: Model of the control loop representing the autonomic computing architecture [64].

The control loops, or managers can communicate with other relevant systems or functions and also with higher level managers. This means that each system or function can govern itself in an autonomic fashion, but can also be monitored by a control and management function that can take any corrective action it sees as necessary. IBM have conceptualised this and have produced the following model based on their autonomic principles.

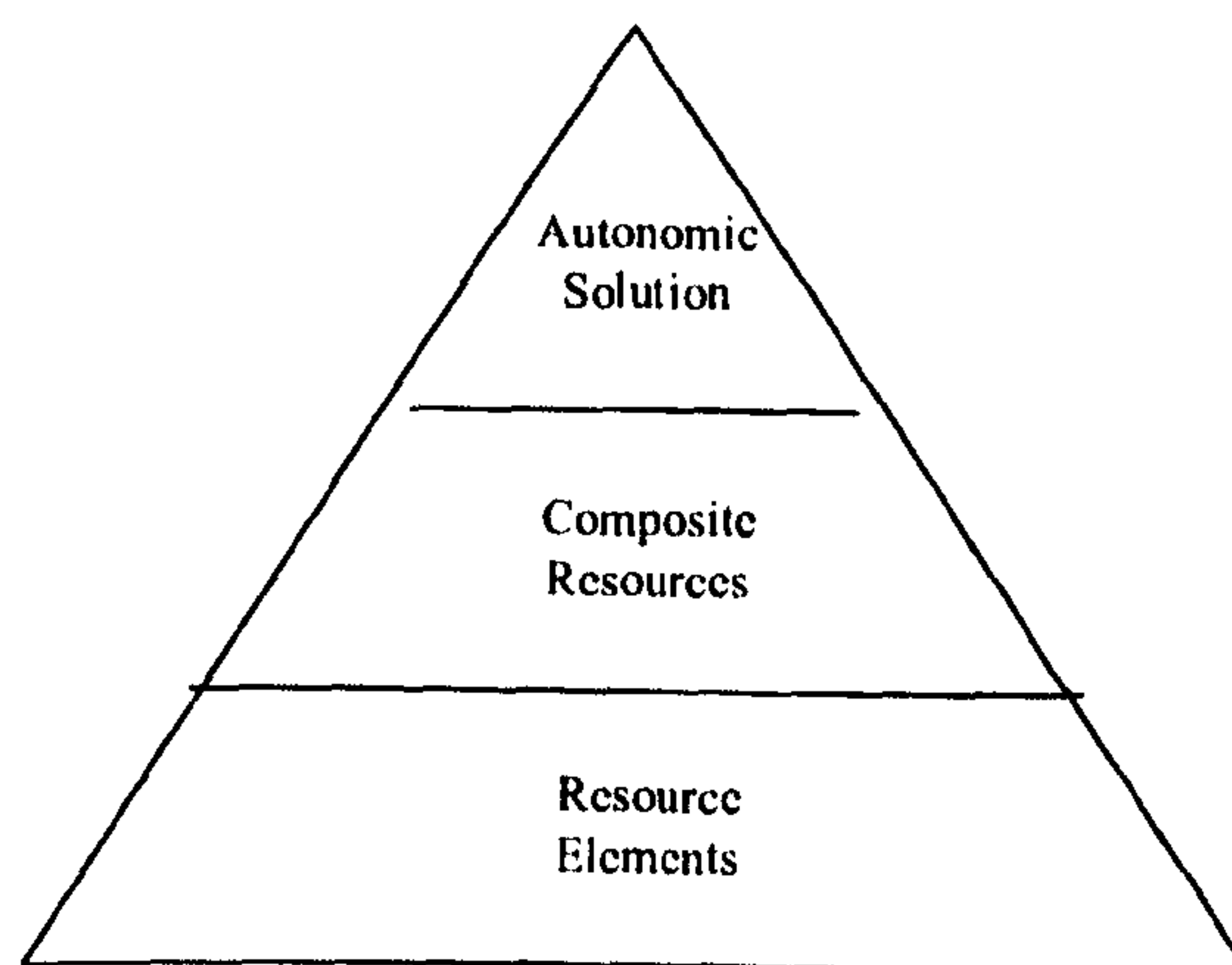


Figure 8: The hierarchy pyramid of the autonomic computing technologies [65].

The model above represents the hierarchy in which autonomic computing systems operate. IBM has developed this three tiered approach, with a more detailed explanation of each level [66]. The lower level of the pyramid consists of the



resource elements of enterprise networks, servers, storage devices, applications, middleware and personal computers. Autonomic computing begins in the resource element level, by enhancing individual components to configure, optimise, heal and protect themselves.

In the middle level of the pyramid, the resource elements are grouped into composite resources, which begin to communicate with each other to create self-managing systems. For example, a pool of servers that work together to dynamically adjust workload and configuration to meet certain performance and availability thresholds can represent this. It can also be represented by a combination of heterogeneous devices such as databases and Web servers that work together to achieve performance and availability targets.

In the highest level of the pyramid, composite resources are tied to autonomic solutions, such as customer care systems or an electronic auction system. True autonomic activity occurs at this level. The solution level requires autonomic policies, schedules, service levels and so on, and drives the consequences of process based optimisation down to the composite resources and even to individual elements.

### **3.4 Chapter Summary**

Recent years has seen a move from computing systems that require administration by humans towards more self-adaptive and autonomic systems. Although this change has started we are still a significant way from such systems being considered commonplace. However, more and more people are beginning to realise the benefits to systems that can control and manage themselves whilst being able to adapt to changes in their operating environment.

The principles and philosophies covered in this chapter have not been embraced into the world of information systems thinking before. However, it can be seen that there are capabilities present within both approaches that would complement a generic framework that is to be used in real world situations to resolve complex and long standing issues.

In this chapter the aim has been to review the background of the principles and definitions required in this research beginning with the notion of self-adaptation and autonomy. This demonstrates what functionality resides within self-adaptation and autonomy, along with what is required to make these approaches work properly. The following chapter provides a literature review into the world of self-adaptive and autonomic computing approaches.

However, as will be discussed in the following chapter, the functionality available overlaps into the world of information systems thinking and can be incorporated into the development of a generic framework. This is particularly important for SSM and VSM, which can both accommodate the functionality outlined in this chapter, as part of their core abilities when combined to make the generic framework. Both methodologies can cater for this functionality when combined and will provide a solid baseline for the generic framework.



## **4. Problem Definition**

### **4.1 Introduction to Problem Definition**

This chapter focuses on research that has already been carried out focused on the world of IT systems and their management and coordination approaches. This chapter does however highlight the problem situation that exists. Whilst extensive effort has been directed towards IT systems and their management and coordination information systems thinking has not been included in this process, which presents itself as information starvation.

This chapter establishes the aim of demonstrating the need for a generic framework to be able to develop a new approach to implement an appropriate solution to this problem situation.

A vast amount of research material exists that focuses and contributes to approaches that are currently employed to support the management and self-managing of software. This includes the capacity for the system to carry out its own monitoring and control. In addition to this system adaptation and its ability to embrace the concepts of self-adaptation and autonomy have also been researched.

The aim of this chapter is to provide a review of current literature that describes emerging practices whilst identifying which approaches are transferable into the world of information systems thinking. The review will be structured into two sections discussing research carried out in the following areas due to the difference in approaches and respective functionalities [67]:

- The static management of software systems; and
- The dynamic management of software systems.

## 4.2 The Static Management Approach

Model based approaches in management include conflict resolution and coordination, strategies and planning. This approach also discusses the potential benefits of incorporating exception handling capabilities into the generic framework.

### 4.2.1 Conflict Resolution

A conflict resolution can be characterised as a process by which a mechanism is started, which is activated as soon as a failure or conflict is detected. In this case Nuseibeh *et al.* have described a failure or conflict in an environment as being:

*“...the situation in which two descriptions do not satisfy some relationship that should hold between them...” [68]*

From the IT system perspective, Nuseibeh *et al.* [69] described the four major steps of failure or conflict management as being:

- Monitoring for failure or conflict;
- Diagnosis;
- Handling; and
- Monitoring the outcome.

Nuseibeh *et al.* state that these four steps can be accommodated into the world of information systems thinking and can become a powerful asset to the generic framework that is to be constructed. These processes provide great flexibility for selecting appropriate resolutions to any failures or conflicts along with coordinating and monitoring the action that has been taken.

According to G. Bieber the coordination functionality is used to ensure that every part of the system [70] acts in accordance to a defined plan or policy. This develops into a capacity that enables, in our case, an information system that is capable of working in its operating environment without its performance taking a



significant downturn if failures or conflicts are detected as stated by Castelfranchi [71].

Research into conflict resolution approaches in IT systems by Castelfranchi has shown that there are four main strategies that are utilised. The concepts put forward in these four approaches can assist the generic framework, and in turn the information system, achieve their aim of being able to cope with changes in their environment. The four strategies for conflict resolutions are as follows:

- Negotiation;
- Arbitration;
- Voting; and
- Independence.

Each of these strategies is reviewed in the following sections in this chapter.

#### **4.2.1.1 The Negotiation Strategy**

The negotiation strategy is frequently used in software systems which have responsibility for supporting conflict resolution processes and are described by Lander and Lesser as the following [72]:

*“...the term used in distributed problem solving research to denote the process by which autonomous nodes coordinate their views of the world and act and interact to achieve their goal...”*

Although Lander and Lesser describe the negotiation strategy of conflict resolutions in an IT context it can be seen that this approach could be adopted into the world of information systems thinking. In this instance the description above could easily be adapted to represent information systems that are managed and coordinated but have the capacity to operate in a self-adaptive and autonomic way whereby they can establish their own resolutions to failures or conflicts that may occur.

The negotiation strategy can be grouped into two main categories according to Cooper and Taleb-Bendiab [73]. Mathematical model based systems are generally based on game theory and economic behaviour. The majority of such developed systems have used quantitative models. The following models, stated by Badr, are considered to be quantitative; multi criteria decision making, conflict analysis, group decision theory, multi objective linear programming and fuzzy arithmetic in searching for an optimal solution based on the negotiation criteria and user preference [74]. Heuristic model based systems have been developed using Artificial Intelligence techniques to support multi agent negotiation behaviour. For instance, as highlighted by Sycara [75] who has developed a 'hybrid' negotiation model, which combined case by case reasoning and a multi attribute utility theory to generate solutions for a given negotiation process. However, this approach does not at present consider the potential impact of change and more specifically, subsequent conflicts emerging as a consequence of the conflict resolution process itself.

Lander and Lesser [76] have suggested that the negotiation strategy is the most appropriate approach to dealing with disparate systems. It can be seen that in particularly complex information systems that this approach would be beneficial in determining the most appropriate resolution to failures or conflicts. Klein [77] has described a protocol that describes how the negotiation strategy works. Klein's protocol for the negotiation strategy is shown below:



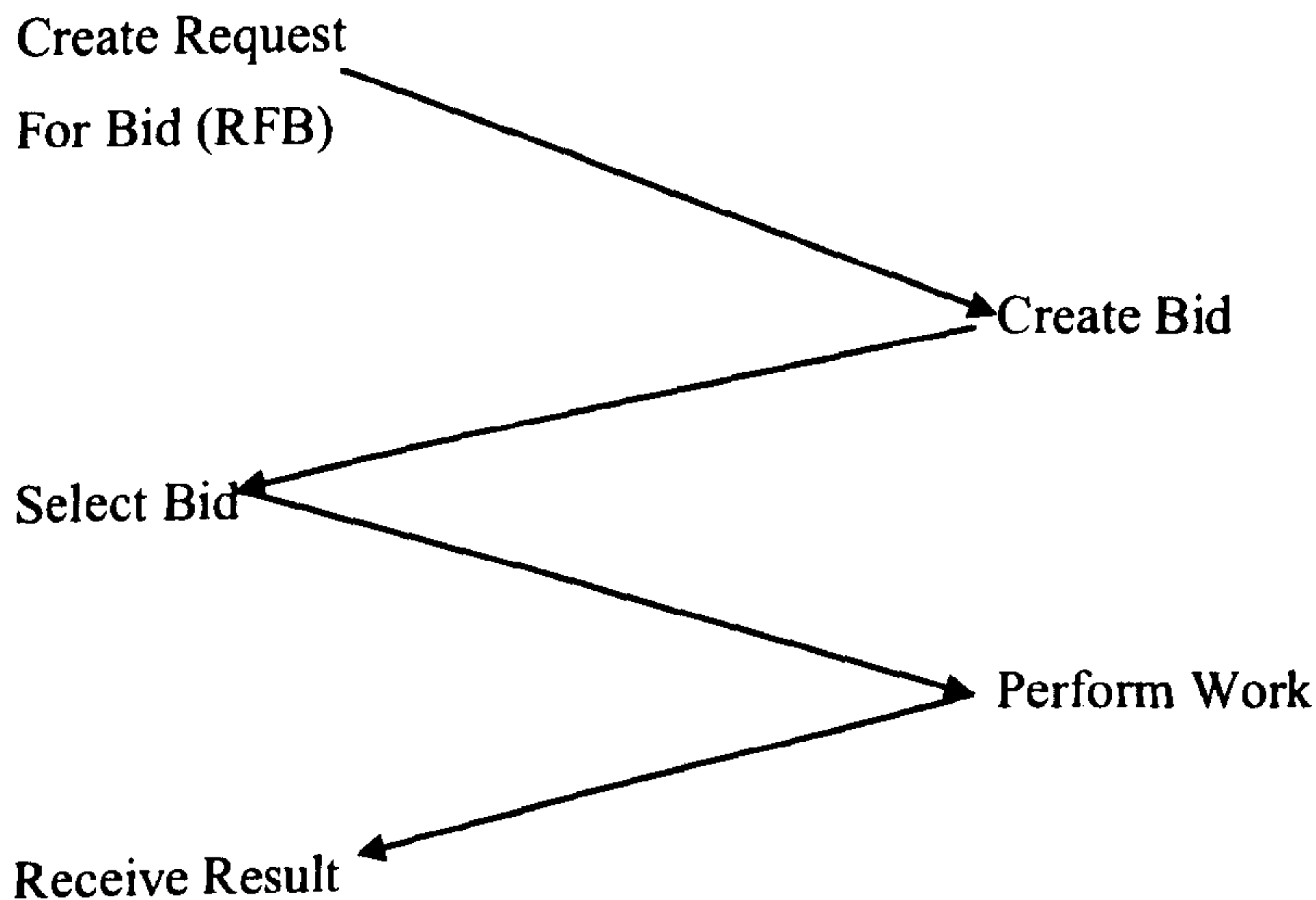
ContractorSubcontractor

Figure 9: A simplified version of Klein's Contract-Net Protocol [78].

The diagram above shows how the process of negotiation can be performed. In Klein's approach a failure or conflict has been detected, which starts a process of negotiation between systems and the management functionality that leads to a resolution being selected and implemented. In the world of information systems thinking the ability to negotiate and select appropriate conflict resolutions can be fully embraced into the generic framework's functionality.

Rosenchein and Zlotkin [79] developed their own eight step model that represents the negotiation strategy and how it supports conflict resolutions. The approach proposed by Rosenchein and Zlotkin is structured as follows:

- Detect conflict: by using a service's constraints for comparing its goals. This compares what should be done with what is actually done.
- Identification of exceptions and select the appropriate handler.
- Negotiation team formation and reformation to identify and agent required for executing a selected plan.
- Solution generation, which generates an ideal solution for conflicts that have arisen.
- Solution evaluator, which evaluates a previously generated solution.

- Negotiation monitor, which is used to gather the required information to support the decisions made by the negotiation control phase.
- Negotiation management and control is a management mechanism to resolve the detected conflict.

Based on Rosenchein and Zlotkin's negotiation model Cooper and Taleb-Bendiab [80] developed a computational framework to support a management and control mechanism for multi agent systems. The computational framework used adopted the negotiation strategy approach to enable the delegation of control of authority between agencies and human users. According to Cooper and Taleb-Bendiab a fully autonomic information system will not work as efficiently as one that still allows the analyst to intervene at certain points and overrule a decision that has been made as information systems are required to deal with people and not just be used as mechanisms to implement IT solutions.

Following on from Cooper and Taleb-Bendiab's approach Ephrati and Rosenchein developed a new model for adjustable autonomy. The control mechanism developed by Ephrati and Rosenchein [81], which forms part of the model for adjustable autonomy, is described as follows:

- Human and computer interface: the interface provided for a user to input control constraints and required information easily.
- Control profiles: a system library of control information containing three key areas:
  - Conflict resolution strategies: which set of strategies and plans are to be used for resolving the conflict.
  - Control gate: this observes and alters the plan when necessary.
  - Control preferences: used to adjust the control profile's behaviour.
- Planning system: the most important module in the mechanism that executes all the tasks in the control mode and contains a script interpreter, matching plan list, plan selection and control resources.
- Control resources: where the plans and strategies are arranged as a system call library, plan library, strategy library and profile library.



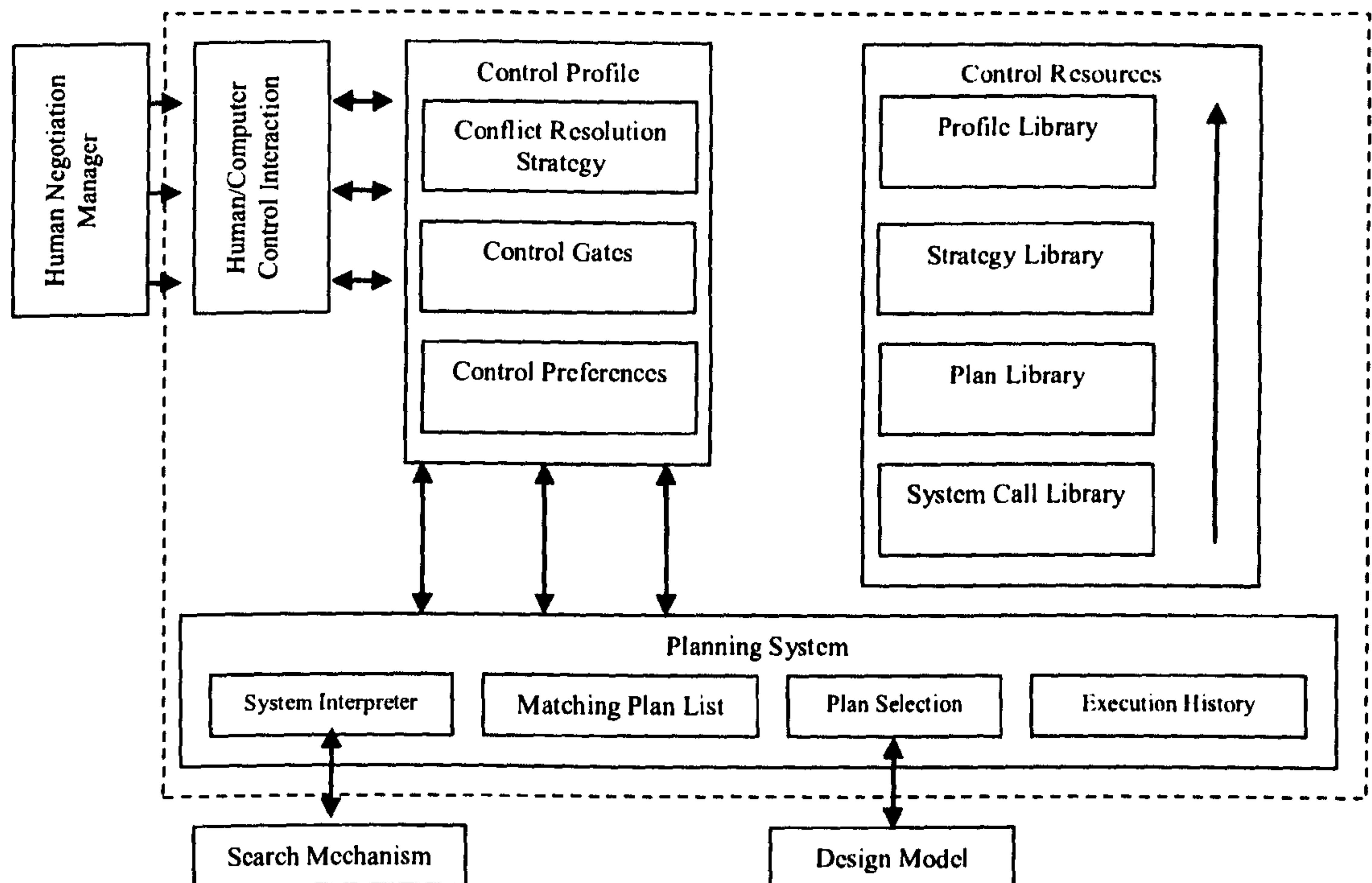


Figure 10: The control mechanism of Ephrati and Rosenchein's model for adjustable autonomy.

#### 4.2.1.2 The Arbitration Strategy

Rosenchein and Zlotkin [82] have defined the arbitration strategy as “...a process in which conflicts are arbitrated by a third party, acting as the arbitrator, who has no direct power to modify the conflict behaviour.” With this approach all information systems that are conflicting with each other must agree on an appropriate corrective course of action.

As is the case with the negotiation strategy the management and control functionality should be in possession of the relevant policies and can help determine what resolution is the best to adopt in a given situation. In many ways the arbitration strategy can be seen as an extension of the negotiation strategy as a general consensus between conflicting systems cannot be reached without some form of communication and negotiation.

The benefits of establishing effective lines of communication through arbitration and negotiation can be of particular benefit to both the generic framework being developed in this study and any information system that is constructed.

#### **4.2.1.3 The Voting Strategy**

Voting is a conflict resolution method adapted from the field of human organisation research, and views authority as being distributed within a society. In the information systems thinking world this relates to information systems having delegated authority to control and manage themselves with regards to general day to day running.

Cooper and Taleb-Bendiab [83] determined that in a distributed system environment there is a need to elect the new agents to assume managerial duties when the previous manager loses its function and cause conflicts. Ephrati and Rosenchein [84] utilised the voting strategy to develop a principle of maximum social welfare, where *“voting for preferences and maximum candidates serves to maximise total satisfaction.”* This strategy was used as a basis for the construction of a plan that brings a society of distributed agents together to achieve a maximal social welfare state in a study conducted by Ephrati and Rosenchein [85].

This approach has some benefits when applied to the world of information systems thinking. For example, the concept of some delegated authority assists the information system to control and manage itself whilst still having the capacity to intervene should any failure or conflict be detected.

#### **4.2.1.4 The Independence Strategy**

Adler *et al.* describe the independence strategy as something that is *“used when a conflict is detected between independent members which are not required to interact with each other to solve their conflicts”* [86]. The independence strategy can



be seen as providing a simple and effective strategy that allows systems to have self-adaptation functionality.

Chang *et al.* [87] describe the independence strategy by using the example of a multi robot system, where there is a collision between two robots. The result of this is that the robots must use their self-adaptation functionality to develop new plans and ensure that a collision does not occur again. This approach is something that the generic framework must consider as the ability to adapt to problem situations, and learn from experience, is crucial to its success.

Lander and Lesser [88] state that the independence strategy approach enables the delegation of given problem solving tasks to independent systems, each with their own individual expertise, through implementing this approach, enables the generation of individual solutions for each assigned system's activities and functions. These are then integrated into an overall solution to the considered problem. The idea put forward by Lander and Lesser is that the application of the independence strategy allows the systems to fully embrace management and coordination processes.

#### **4.2.1.5 The Four Strategies of Conflict Resolution**

It can be seen that each of the four approaches discussed above can be considered valid for the development of a generic framework. Indeed, such a generic framework would not be able to fully embrace the concepts of self-adaptation and autonomy if it was to overlook any of these strategies. It is important that the generic framework is constructed in such a way that it is able to select which approach is the most appropriate for any given problem situation and implement it.

At the core of each of these strategies is the capacity, in one form or another, to delegate certain amounts of authority to activities or functions within information systems. This provides the basis for control, coordination and management functionality to be incorporated into the design of the generic framework.

### 4.2.2 Strategy and Planning

Barber *et al.* [89] define the strategy and planning approach as:

*“A strategy is an abstraction that the agent can use to encapsulate the coordination mechanism used for any of the core problem solving tasks...”*

In terms of the creation of the generic framework this relates to implementing a strategy that is able to understand the problem situation and select the most appropriate resolution. Barber *et al.* have developed a model for conflict resolution strategies. The model is initially informed of a failure or conflict and then looks to employ a conflict resolution. Before a conflict resolution can be implemented a set of requirements must be met according to Barber *et al.* [90]. The strategy requirements may make use of different parts, or may also place constraints on the reasoning capabilities.

The execution of each strategy differs according to different resources that have been used. Barber gives the example that some strategies may require a larger number of messages or a longer time. It is important to consider this factor when dealing with deadlines or limited resources.

The solution quality provides the possibility of multiple solutions and therefore each solution has different qualities depending on the use of different strategies.

Domain requirements ensure that systems domains should also be considered and satisfied by the proposed solution strategy.

The concepts of strategy and planning will play an important role in the generic framework as it develops information systems that are able to implement mechanisms that will lead to the selection of resolutions.



### 4.2.3 Exception Handling

An exception has been described by Klein *et al.* as:

*“...any deviation from an ideal collaborative process that uses the available resources to achieve the task requirements in an optimal way [91].”*

Exception handling has existed for the past couple of decades and has worked effectively throughout that time. However, over the last fifteen years or so there has been a renewed interest in exception handling to support self-adaptive software with modifying code behaviour. This work has focused on the development of dynamic exception handling IT systems. However, the concepts of exception handling can be translated into the context of information systems thinking and can work as an extension of the strategy and planning functionality of a generic framework.

Visser [92] has proposed an exception handling framework, which adopts a high level management and control system design to monitor a system's execution information model and trigger exception handling processes should a failure or conflict be detected. As highlighted by Visser, in a framework such as this, the exception handling concept utilises the application in three models. The monitoring model represents any mismatch between the system's normal or required system state with the current system state. The clarification model contains exception signatures with their associated class or type. The verification and recovery model inspects system features in order to recover the system once again.

Klein *et al.* [93] addressed the management and control of exception handling processes by developing a category of exceptions and a mapping to appropriate resolutions. These processes have been called the 'Exception Management Meta-Process' and are demonstrated in the model below:

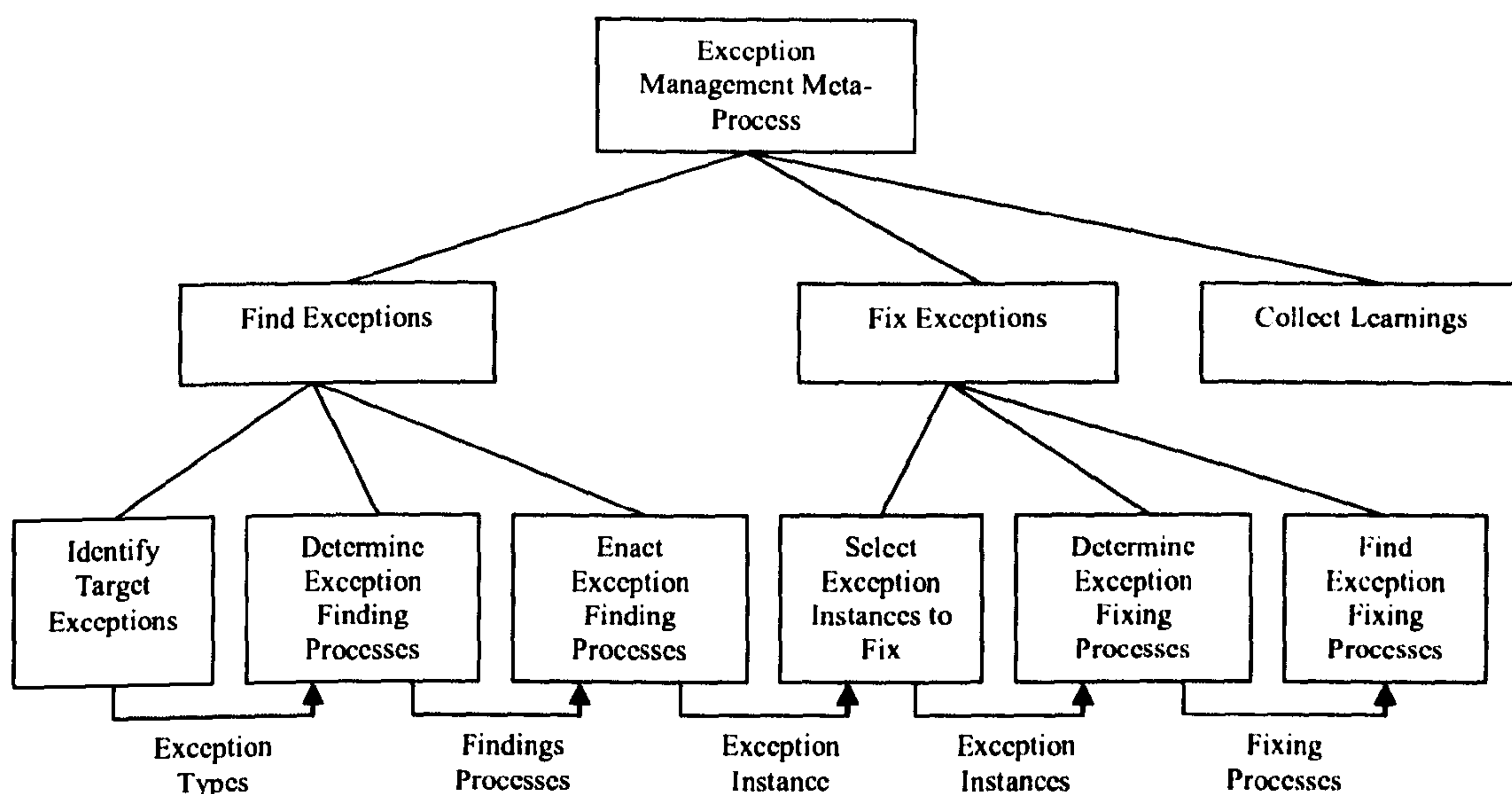


Figure 11: The generic 'Exception Management Meta-Process' model by Klein and Dellarocas [94].

The 'Exception Management Meta-Process' is a high level exception handling framework which can be adapted to suit the generic framework as the concepts put forward here complement the world of information systems.

When exceptions are detected checks for the cause of the failure or conflict are sought, as discussed by Klein. Once the root of the problem situation has been defined exception handler processes are enabled, which leads to the selection of an appropriate resolution. A model developed by Klein and Dellarocas, shown below, summarises the exception handling process:

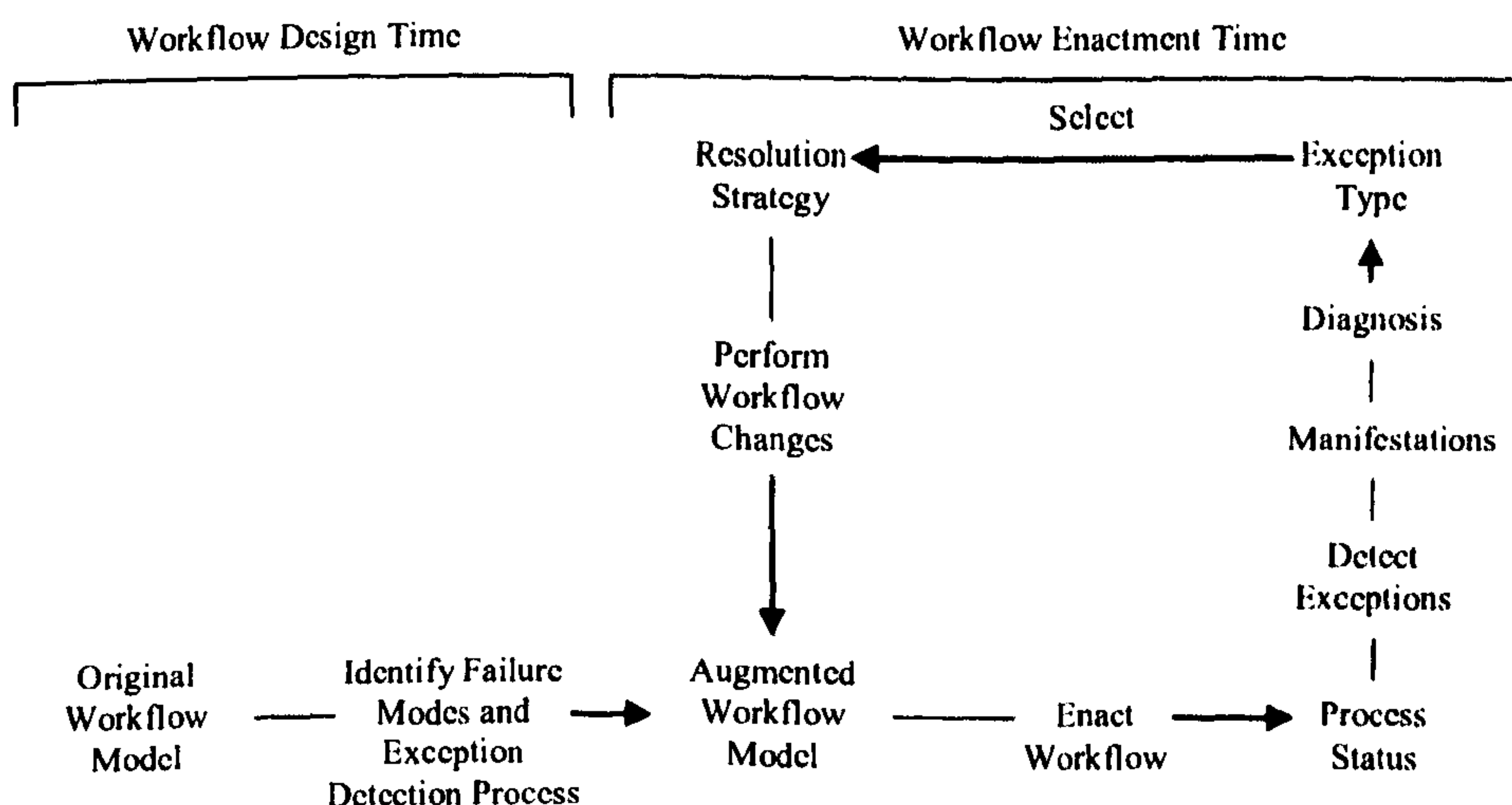


Figure 12: Model which summarises the exception handling process by Klein and Dellarocas [95].



The autonomic computing vision of IBM [96] is based around the notion of systems self-managing and controlling and requires individual systems to conduct *“self-monitoring, self-diagnosing, self-managing and self-adaptation.”*

Oreizy and Taylor discuss the management, adaptation and reconfiguration of systems [97]. It should be pointed out here that there is an overlap in terms of autonomic exception handling concepts and those that will need to be incorporated into the generic framework.

One aspect that cannot be overlooked is that Oreizy and Taylor did not consider the assurance, predictability and the impact of any proposed system adaptation on the system as a whole. This could potentially lead to undesirable system states and even more failures or conflicts than there were before if not properly managed and controlled. Therefore, as will be discussed in the following chapters, it is essential to develop a generic framework that considers the consequences of any such information system adaptations.

#### **4.2.4 The Deliberative Approach**

Dastani et al. [98] have conducted extensive research into the communication between systems, which has led to the forming of various theories in relation to deliberative systems architectures. The most commonly known and acknowledged model for a deliberative architecture is the ‘Beliefs, Desires and Intentions’ model (BDI), reviewed by Rao and Georgeff [99], and which was first introduced to the world of computing by Bratman *et al.* [100]. According to Bratman *et al.* BDI *“demonstrates the representations of aims, beliefs and desires about its environment to give suitable responses to be autonomously determined, planned and executed”* [101]. The definition of aims, beliefs and desires about internal and external environments is a philosophy that can underpin the generic framework, or any approach in the information systems thinking world for that matter.

In addition to this the BDI Model can be incorporated into a management and control structure as it contains functionality which can store details on beliefs and

desires along with any relevant policies and conflict resolutions. Bratman, whose model is shown below, states:

*“...the belief operator  $B$  denotes possible plans, the goal operator  $G$  denotes relevant plans, and the intention operator  $I$  denotes plans the agent [system] is committed to.”*

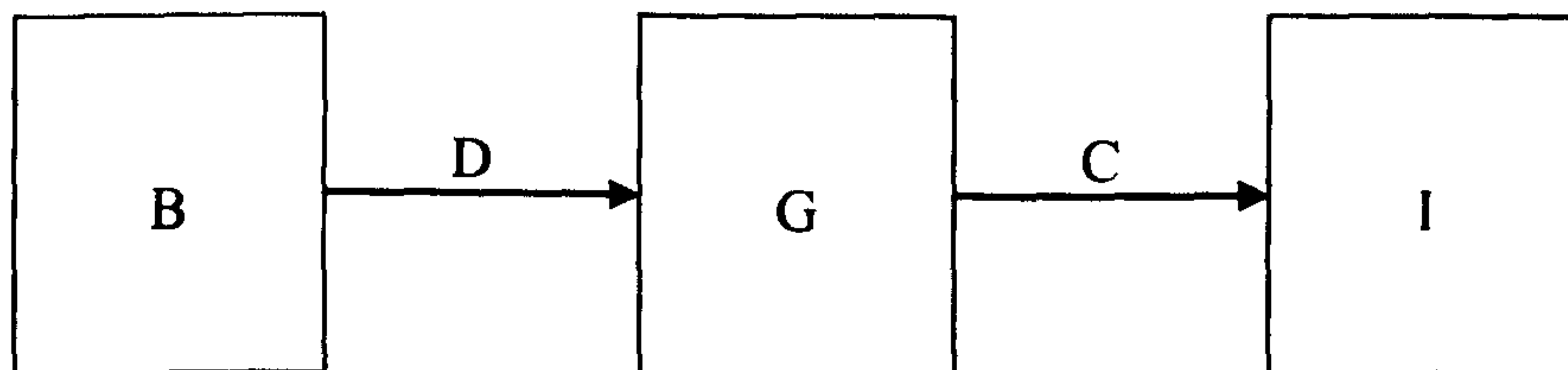


Figure 13: Relations between beliefs, goals and intentions as stated by Bratman [102].

#### 4.2.4.1 Extensions to the BDI Model

Many extensions to the BDI Model were proposed soon after its development. Bratman further expanded the BDI by developing the ‘Intelligent Resource-Bounded Machine Architecture’ (known as IRMA). IRMA approach incorporated library which stored the beliefs, desires and intentions of the relevant system. This approach included a reasoning process so that it could effectively communicate with its environment.

Through adopting the functionality put forward by the BDI Model the generic framework will be able to perform analysis of the current environment and determine the difference between the current system state and the desired system state. This would then lead to a filtering process responsible for determining the most appropriate resolution to achieve the desired state. Following on from this a deliberation process is started that considers the recommended options for resolving the problem situation and selects the one it considers most appropriate.



#### 4.2.4.2 The Epistemic Deontic Axiologic Model

The BDI Model, as discussed in the previous section of this chapter, has had many extensions to it proposed. Filipe states that *“the main shortcomings of the model are that it lacks support for revised beliefs, desires and intention sets”* [103], whilst Castelfranchi *et al.* add that *“there is a lack of support for situated and normative intentions”* [104].

Castelfranchi *et al.* suggest that the BDI Model does not store any policies or strategies within its management and controlling functionality and as such it does not cater for system attitudes or reasoning associated with a set of intentions, which have to be considered to select the correct decision. It is discussed that although the BDI Model is capable of organising the resolution to a problem situation it is not capable of conducting further changes. In the world of self-adaptive and autonomic information systems thinking it is crucial that any approach adopted can cope with constant changes in both the internal and external environments according to IBM [105].

Laws *et al.* [106], proposed a new architecture for self-adaptive software, based on both the Bratman's IRMA and the Viable System Model by Beer [107], where functionality has been incorporated into the model to compensate for the deficiency they call *“blind intentions.”* The approach that Laws *et al.* put forward would complement the generic framework that will be developed as the reduction of blind intentions is one of the first steps towards achieving true self-adaptation.

Filipe [108] introduced the Epistemic Deontic Axiologic (EDA) model, which is a normative based model for distributed systems that incorporates social activities along with obligations and their norms. In this context, Castelfranchi *et al.* [109] describe norms as being:

*“...simply built-in constraints in the system's architecture or rules and protocols the systems necessarily applies...”*

The EDA model is developed by Felipe with three core functionalities, which are:

- A cognitive, epistemic component to adopt the degrees of beliefs or disbeliefs, formalising their plans and procedural abilities.
- A behavioural, deontic component that predisposes the system to respond with actions and plans to social obligations.
- An evaluation axiological component that contains the order of the system's preferences when considering its norms.

The EDA model contains a deontic component that represents itself as strategies and plans and is viewed as either social obligations or their norms, which have been defined above. Therefore, as Felipe states that *"deontic norms determine and evaluate the system's behaviour based on the epistemic state by considering the axiological norms for resolving arisen conflicts."*

Considerable effort has been spent working towards developing a fully self-adaptive extension to the BDI Model. Although this work is ongoing it can be seen that the general philosophy that underpins the BDI Model is something that fits well within the boundaries of a management or control function that must exist within the generic framework, which will also help achieve true self-adaptation and autonomy.

The BDI Model will be included in the generic framework's functionality due to its inherent ability to support self-adaptation and autonomy. However, the IRMA and Epistemic Deontic Axiologic Models will not be included in the generic framework. This is due to SSM and VSM already containing the additional functionality supported by the IRMA and Epistemic Deontic Axiologic extensions of the BDI Model.

### **4.3 The Dynamic Management Approach**

This section of the chapter looks at dynamic approaches to systems management including policy based management, event based management,



architecture based management. This approach also discusses the potential benefits of incorporating autonomic management capabilities into the generic framework.

#### 4.3.1 Dynamic Policy Based Management

Significant research has already taken place looking into the use of dynamic management for large systems, which can be applied to large organisations in the world of information systems thinking. Moffett *et al.* [110] stated the importance of being able to dynamically update policies for the managing of systems, where policies are defined as follows:

*“...the plans of an organisation to achieve its objectives...”*

The concept of policy management, which includes action and conflict resolution strategy policies, is represented as a policy hierarchy, where each policy in the hierarchy represents a strategy that meets a defined aim or objective. In terms of the generic framework this establishes the boundaries, activities and conflict resolutions for information systems.

At this point it is worth acknowledging that Sloman [111] states that *“the separation of policy management from the policy interpreter (such as managers) facilitates both the dynamic change in the system management process and the reuse of managers in different processes”*.

It is a prerequisite for all system functionality, conflict resolutions, aims and objectives to be coordinated. Sloman [112] studied levels of delegation and authorisation for programmable computer networks that would ensure that should a failure or conflict be detected that it would be able to implement suitable *“...event triggered rules which can perform actions as network components...”* according to Bieber and Architect [113].

This approach, which will also be taken into consideration for the generic framework, ensures that under certain circumstances, where boundaries have been

predefined, the appropriate resolution can be selected and implemented by the information system in an autonomous fashion. However, as Sloman also states, this can lead potentially lead to further failures or conflicts that had not been experienced prior to the resolution being selected and implemented. An example of this is as follows:

*"...an obligation policy may define an activity which is forbidden by a negative authorisation policy; there may be two authorisation policies which permit and forbid an activity or two policies permitting the same manager to sign cheques and approve payments."* Sloman [114]

Sloman [115] discusses that policies management is an important functionality in terms of the generic framework but requires constant policy specifications updates and well defined conflict resolution strategies. Sloman *et al.* [116] determine that policies can be grouped though similarities in problem situations or by the information systems that would carry out resolutions. Sloman *et al.* [117] further developed this approach by introducing the 'Ponder' language, which has the capacity for specifying a set of policies according to their type. The five main types are as follows:

- Authentication policy;
- Obligation policy;
- Policy groups;
- Roles; and
- Domains.

Sloman *et al.* [118], since establishing the five types of policy as shown above, have proposed a framework which is developed to provide *"automated policy deployment and flexible event triggers to permit dynamic policy configuration, focusing on solutions for dynamic adaptation of policy in response to changes within the managed environment."*



The framework developed by Sloman *et al.* consists of two types of policy adaptation, which are as follows:

- Dynamically changing policy parameters to set new attribute values of the managed object.
- Reconfiguring the policy objects by selecting, enabling, disabling or load predefined quality of service policies that are stored within a management or control functionality.

Both Barber *et al.* [119] and Verma *et al.* [120] propose that quality of service and service level agreements should be incorporated into dynamic policy based management. It should always be remembered that in both IT systems and the world of information systems thinking that policy failures could be possible and should be expected in any environment. As a consequence, a validation process is still required to support such self-adaptive management approaches. In terms of the generic framework this means that not only will the information system require constant monitoring but also the management and control functionality itself.

Yoshihara *et al.* [121] have developed a new approach that adapts policy parameters whilst being strictly monitored. This framework utilises a management capacity to orchestrate monitoring and compliance checks with the assistance of a policy working group. The policy working group aims to establish five key levels of service and sets definitions and boundaries that are implemented during monitoring checks. These five key levels are as follows:

- Policy specifications;
- Policy management;
- Life cycle;
- System's notification related to quality of service threshold violations; and
- Prototypes using differentiated systems.

Brunner *et al.* [122] have looked to further develop the framework Yoshihara *et al.* have proposed their own system that is designed "*for managing the quality of*

*service in Multi-Protocol Label Switching (MPLS) networks by extending the Common Interface Model (CIM) policy model with MPLS specific classes."*

Beardon *et al.* [123] have, in a similar fashion to Brunner *et al.* suggested a new model could be implemented. The Policy Core Information Model (PCIM), which they discuss is to be utilised for *"supporting goals by using monitoring data to evaluate whether the specific goals are satisfied or not."* Referring to Sloman *et al.* [124] the goals defined by Beardon *et al.* are to be considered terms of obligation policy rules.

It can be seen in this section that significant research has been conducted into dynamic policy based management, with all of the concepts put forward relevant to the development of the generic framework. Whichever methodologies were to be chosen for the generic framework it is expected that the concepts and philosophies discussed here are to be within the capabilities of those methodologies. With SSM and VSM being the chosen methodologies it is seen as being a prerequisite functionality and one that will fully support self-adaptation and autonomy.

#### **4.3.2 Dynamic Event Based Management**

Large scale organisations and information systems would benefit from effective dynamic event based management. Extensive research in this field has recently seen the development of dynamic event based approaches by Banavar *et al.* [125] and Carzaniga *et al.* [126], in which events are the basic communication mechanisms. In terms of the generic framework this functionality will end up playing a crucial role. Cugola *et al.* [127] explain how the dynamic event based approach works in the following way:

*"The process starts with event 'subscribers', where a system is registering an interest in receiving a particular event's notification, then event 'publishers' respond to the previous subscription by publishing those subscribed events to all event subscribers. Therefore, this model solves the communication problem between subscribers and publishers."*



Bacon *et al.* [128] further develop the concepts put forward by the subscriber and publisher system by establishing the type of event that was currently in place. The integration approach introduced deals with an event as an object class and classifies them according to their type. At this point the subscribers highlight their interest by subscribing to the type of object classes that they have become interested in.

With this approach it is difficult to optimise performance or distribution as it requires efficient implementation to begin with and classifying types can be a complicated process. However, if this can be achieved the dynamic event based approach is certainly beneficial to both the IT solution and information system thinking worlds.

There are many new approaches that are just being introduced, which are detailed below. All of these approaches, as is the case for the BDI Model, look to further develop the research that has been carried out before and take dynamic event based management on to the next step.

Siena, developed by Carzinga *et al.*, [129] describes itself as a “*distributed subscribe and publish content based system, consisting of network event brokers and focuses on a global broadcast operation for advertising and disseminating through the network.*” Siena, however, is a static management approach and, as a consequence, is unable to deal with any failures or conflicts that have been detected.

JEDI, developed by Cugola and Nitto [130], is a Java based event approach that utilises the functionality of the Java programming language to group types and classes into a service framework. The objects have similar behaviours and are implemented to deal with failures or conflicts that occur. Although JEDI focuses on disconnect and reconnect operations only it is considered that it can be expanded further into the information systems thinking world with extended functionality, as is the case with Bacon *et al.* and their proposed approach. One drawback, however, is that JEDI does not contain fault tolerance functionality.

Cabrera *et al.* [131] introduced Herald, which is an event based notification framework for a subscriber and publisher service architecture, further developing the

concepts put forward by Bacon *et al.* (2000), using subscribers, publishers and establishing appropriate positions for the two to meet. This approach gives the framework the opportunity and functionality to deal with more systems than other event based notification approaches and has the additional ability to explore dynamic system reconfiguration.

Both Bayeux [132] and Scribe [133] have developed what they call topic based event approaches that use communication channels that operate by routing a message to the identified topic. In the case of the generic framework the communication channels are crucial to its success in achieving self-adaptation and autonomy. However, the communication channels are used for dealing with fault tolerance functionality. This means that all the events that occur must still be communicated by the communication channels, which can become a limitation.

Pietzuch *et al.* developed the Hermes approach [134], which is a *"distributed event based middleware approach that attempts to address most shortcomings attributed to type based and attribute based subscriber and publisher models by focusing on event types and then filtering within the event attributes."*

Pietzuch *et al.* [135] have introduced an enhanced framework based on Hermes, which considers event detection as its most important functionality. Shanahan [136] states that this could be enhanced further through the implementation of an additional feature that could potentially be added to the event by using event calculus. This would create a situation within the system that is triggered by the detected event with a defined time duration.

Friday *et al.* [137] have suggested that policies used in dynamic event based management requires flexible policies to ensure that system adaptation and coordination according to system requests can be maintained. This approach is something that can be derived into the information systems thinking world, as its functionality would complement current methodological approaches. Friday *et al.* develop this approach further by introducing the concept of generic models and applying them to subscriber and publisher systems in a self-adaptive and autonomic



fashion to gather input from the user and establish a reasoning mechanism for the environment within which it exists. This approach is proposed by Garlan *et al.* [138].

### 4.3.3 Dynamic Architecture Based Management

Recent advances in software architecture and description languages have provided new and innovative ways to embrace the concept of dynamic architecture based management in IT systems. As Jones [139] states this work is centred on *“software architecture modelling, analysis and reasoning to support runtime software management, which incorporates adaptation and evolution.”*

In this section of the chapter, the literature review is focused on research concerns relevant dynamic architecture based management approaches and their ability to carry out unplanned architectural model changes. This research can be translated into the world of information systems thinking and the generic framework would benefit from having the capacity to deal with unexpected changes that are required. Changing system architecture also plays a significant role in the system configuration, and reconfiguration, since it enables the system to select the most appropriate resolution. Oreizy *et al.* [140] state that this approach is based on the application of specific policies and requirements and as such allows mechanisms to be put in place that can be reused whenever failures or conflicts are detected. This approach is considered to be a core functionality of the generic framework with the information system being able to call upon such mechanisms to configure or reconfigure itself when experiencing problem situations.

Le Metayer [141] describes software architecture as *“a graph for software architecture formalisation.”* Metayer’s software architecture approach represents the individual systems with well defined interactions and communications. Le Metayer’s architecture ensures that the systems can communicate only along the links specified by the architecture itself. As is the case with the generic framework the communication functionality will ensure that all information systems will be able to communicate with each other.

According to Wermelinger *et al.* [142] develop Le Metayer's approach by suggesting that:

*“For each architecture style, there is a class of architectures specified by a graph grammar. The class formalises a set of architectures sharing common communication models and rules. The rules of the coordinator are checked to ensure that constraints are preserved by the architectural style. Modelling architectures through categorical diagrams and dynamic reconfiguration could be realised by algebraic graph rewriting and describing architectures and operating changes over a configuration, such as adding, removing or substituting components of interconnections.”*

Garlan [143] also adopts the approach suggested by Wermelinger *et al.* and discusses the three main aspects of the architecture, which are as follows:

- Description;
- Constraints; and
- Modification.

Garlan considers the three aspects highlighted above to be the core functionality of the architecture and that it enables a fully self-adaptive control and repair framework. In terms of the generic framework the functionality discussed here, and in the following chapters, discuss the steps that can be applied to the world of information systems thinking. Garlan *et al.* [144] and Eracar *et al.* [145] extend the architecture to consider the feedback and feedforward loops that form part of the communication functionality. This capability enables the architecture to establish change requirements and therefore to fully embrace self-adaptation.

Oriezy *et al.* [146] extend the self-adaptive architecture above by stating that any system, or in this case information system, will need to include functionality that can implement adaptation management, monitor observations, plan changes and deploy change descriptions, or conflict resolution strategies to deal with system management concerns. Andrade [147] supports this by suggesting that the approach



introduced by Oreizy *et al.* *“achieves separation between computation and coordination concerns as a way of providing a higher level of self-adaptability through system reconfiguration.”*

Buyya *et al.* [148] have developed a dynamic architecture based approach to resource management which they believe assist a system achieve its objective of becoming self-adaptive. This functionality, whereby system resources are controlled and management by the information system itself, enables conflict resolutions to establish which strategy is the most appropriate to implement within any given problem situation.

One key aspect when large organisations are required to adapt to changing environments is the level of resources currently available. This is the same for information systems too and the generic framework will need to be able to incorporate information relating to the availability of resources within its own environment.

#### **4.3.4 Autonomic Based Management**

IBM [149] have set themselves a mission statement that *“The systems community needs to design and build computing systems capable of running themselves, adjusting to unpredictable changes and handling resources efficiently.”*

Research has shown that the two main elements of autonomic based management include the functional unit that performs the main operations and the management unit which assumes responsibility for system resources and operation performance and as, stated by Chess *et al.* [150], the reconfiguration of resources according to adaptive changes.

Autonomic systems have been defined by IBM [151] as systems that have:

*“...the ability to manage themselves and dynamically adapts to change in accordance with policies and objectives, which are self-diagnosing and self-healing systems, so these systems have the abilities to detect quality of service and*

*performance, and allow users to accomplish what they request rather than try to handle and repair these computing systems...*"

In recent years the research community has been debating the approaches adopted by autonomic computing and proactive computing. Want *et al.* [152] describe the difference between the two as being "*autonomic computing focuses on managing the computer system complexity but the proactive computing approach adds the need to monitor and build complex real world interactions.*"

Appavoo *et al.* [153] discuss that since autonomic computing systems have functionality that enables monitoring, failure or conflict detection and implement resolution strategies it is considered that they must also be able to dynamically reconfigure themselves. Appavoo *et al.* further suggest that autonomic computing systems should be able to replace or modify their own programming code. In terms of information systems thinking this should be where an information system is able to redefine itself or update its policies, beliefs and desires to cope with changing environments. By embracing this autonomic computing principles the generic framework will be able effectively support itself as it becomes possible to monitor, diagnose and manage the information system. Appavoo *et al.* [154] state that this functionality is particularly useful in systems where each resource is carrying out different functions and activities.

Chess *et al.* [155] expand this approach and state that "*successful autonomic systems not only need to self-detect, self-diagnose and self-heal but to also self-protect to allow autonomic management in a secure environment.*"

These functionalities are taken into consideration when developing generic framework requirements to ensure that the concepts of self-adaptation and autonomy are fully incorporated and supported.



## 4.4 Chapter Summary

This chapter has discussed research that has been conducted in the field of self-adaptation and autonomy and how these findings can be translated into the world of information systems thinking. By their own definition self-adaptive and autonomic systems must be able to manage themselves and adapt to change, but still retaining the capability for human interaction to override any decisions that have been made.

Diakov *et al.* [156] state that large systems cannot be developed to embrace self-adaptive and autonomic capabilities in an ad hoc manner as a structured approach needs to be employed. It is imperative that information systems looking to incorporate such functionality adopt the philosophies discussed in this chapter and ensure that they are taken into consideration when operating in the real world with real world problem situations.

In this chapter the aim has been to provide a detailed literature review of self-adaptive and autonomic computing concepts and requirements in real world situations. The following chapter provides an overview of requirements for the generic framework that will be developed as part of this research project. By assessing how the framework will need to identify conflicts, failures and implement resolution strategies a list of requirements will be created.

The functionalities discussed in this chapter have been reviewed for their inclusion or exclusion based against their ability to support self-adaptation and autonomy during the development of the generic framework.

## **5. Problem Solution**

### **5.1 Introduction to the Problem Solution**

This chapter focuses on research that has already been carried out focused on the world of information systems. From the viewpoint of information systems thinking a generic framework can be seen as the implementation of several frameworks, or methodological approaches, that interact with each other through identification and control methods. Since the early 1980s a significant amount of research has been conducted into self-adaptive systems design and development, which has led to an understanding of the requirements for conflict detection and resolution. Although these concepts have been used in an IT application sense the underlying philosophies can be incorporated into information systems thinking.

This chapter discusses the requirements that will be incorporated into the generic framework so as to ensure that it is able to cope with the demands placed on it in real world situations. The requirements covered in this chapter will allow the generic framework to apply self-adaptation and autonomic concepts from an information systems thinking perspective. This chapter provides four key requirements for the generic framework.

Conflict detection is a monitoring function that has the responsibility for detecting changes either in the generic framework's working environment or within the framework itself.

Conflict identification looks at identifying the conflict and establishing the classification of the failure or change that has taken place. Having achieved this the conflict identification phase will start to determine what processes must be in place to ensure that an appropriate resolution is implemented.

Conflict resolution will be incorporated into the generic framework to ensure that the appropriate resolutions are implemented.



Dynamic adaptation will enable the generic framework to adapt to changes within its operating environment in a self-adaptive and autonomous fashion.

## 5.2 Autonomic Control Requirements

The requirements for developing the generic framework start by ensuring that it has the capacity to detect, classify, fix and reconfigure itself without having to stop whilst it waits for changes to be made manually. By their very nature conflicts are difficult to predict, which requires a new approach to be developed to ensure that the generic framework is able to detect and rectify problems as they arise through embracing the concepts of self-adaptation and autonomy.

B. Nuseibeh *et al.* [157] define inconsistencies as being “...*any situation in which two descriptions obey some relationship that is prescribed to hold between them.*” Essentially this means that an inconsistency occurs when changes or failures in the operating environment are detected. If such an inconsistency is occurred the control functionality of the generic framework must be called into action.

Additionally the framework monitoring functionality will be required to detect any conflicts and, as a consequence, focus its attention and efforts on problem areas within the environment, or framework itself. This capacity will encourage the generic framework to learn from previous problems encountered and act as a Validation and Verification function to ensure that suitable analyses and resolutions are implemented.

In the case of the generic framework the control functionality will conduct analysis based on the information provided by its monitoring capacity. Once the analysis is completed, having understood the various policies and regulations that exist within its environment, the framework will look to implement the most suitable resolution. Whilst this process is being carried out the framework must keep monitoring itself to ensure that the resolution being implemented solves the current problem and checks to see whether the action taken produces any new conflicts or failures. As a result of this the framework must conduct this work continuously to

ensure that it is able to resolve its problems and therefore meet the needs of the environment within which it is operating.

The autonomic control requirements will enable the generic framework to work with self-adaptive, autonomic, self-diagnosing, self-repairing and self-reconfiguring functionality.

### **5.2.1 Detection of Conflicts**

The detection of conflicts, failures or inconsistencies, is the first step towards achieving autonomic control within the generic framework. To achieve this a set of rules and boundaries within which the monitoring function will operate need to be established. To this end it is imperative that from the outset the generic framework must define rules and boundaries before commencing with detecting conflicts.

The generic framework will require rules that determine adherence to its initial aims and environment, establishing what is to be considered acceptable behaviour and any other regulations that are imposed upon it. In a framework such as this it is possible that its operating environment will contain many different strands, so the capacity to deal with problems in this context in a self-adaptive and autonomous way within the constraints of the rules is considered a requirement. B. Nuseibeh *et al.* have established a set of five consistency rules.

In the first instance is the notation's definition that exists in the development process. For example, each regulation in the environment should be consistent with its declaration.

The method's development must consist of a set of notations with some guidelines. In an environment where there are many different strands it is important that definitions are consistent amongst all parties involved.

The model of the development process must define evolutionary steps, the entry and exit conditions for those steps and the constraints associated with each step.



The possible occurrence of a consistency relationship between instances of two objects must be included, even if it has not been determined previously.

There must be consistency during the developmental process upon discovery, specifying and refining of rules and implementing resolutions.

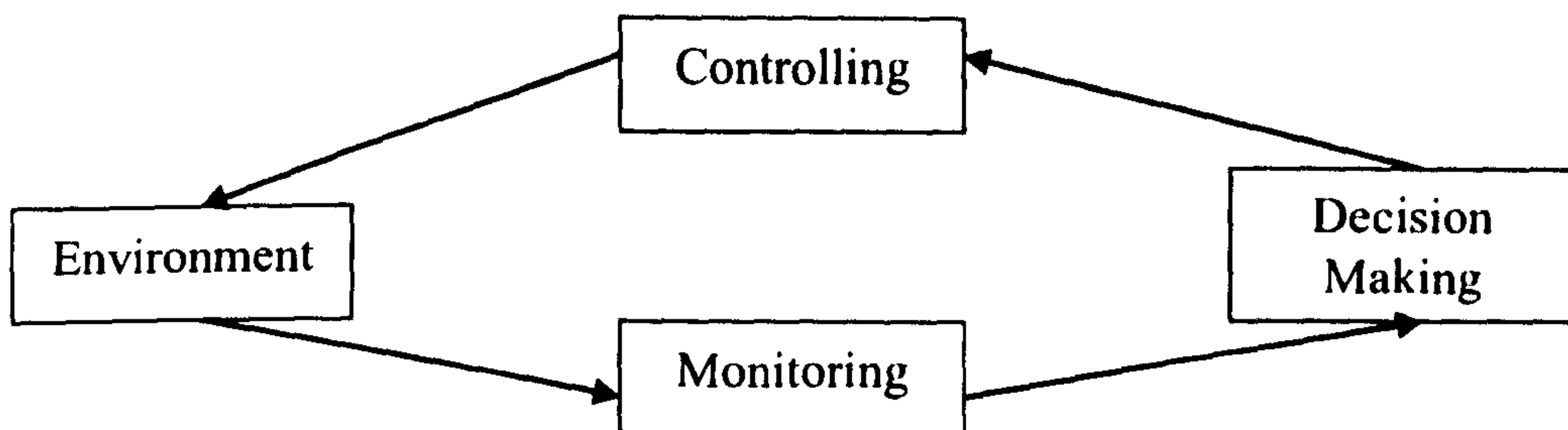


Figure 14: The relationship between management and monitoring as defined by Nuseibeh *et al.* [158].

The monitoring function assumes responsibility for gathering information on performance and interaction with the environment. This process of gathering information must operate within the defined boundaries and will be used to detect any conflicts that may exist. The control process depends on the information provided from the monitoring process as shown above, which shows the relationship that occurs between monitoring and controlling or management functions.

Whenever an inconsistency or conflict is detected an analysis is carried out to try and identify the type of conflict that has been found. It is impossible to determine every eventuality, even though extensive rules and boundaries may be defined, and as such it is imperative that the generic framework is able to develop dynamic resolutions.

That said, conflict detection using controlling regulations can be performed dynamically for a set of policies, rules and boundaries that are already in existence. The monitoring functionality will be utilised to act as a filter and prevent any activities that must not be performed or are not permitted within the environment. However, it should be noted that this functionality is required to detect conflicts that exist and not those that may potentially exist.

### 5.2.2 Identification of Conflicts

The identification of a conflict commences as soon as a detection has been made. The process of analysis and diagnosis according to B. Nuseibeh *et al.* [159] locating the failure or conflict by identifying an inconsistency event and notifying the controller by providing information on the event description and the rule or regulation that was violated. This also extends to determining the cause of failure or conflict. Classification of the type of failure or conflict, which provides the basis for the selection of a conflict resolution strategy, must be incorporated.

### 5.2.3 Classification of Failure and Conflict

By determining the classification, or type, of failure or conflict the generic framework will be in a better position to decide which resolution to implement. There are three broad classifications of failures as highlighted by Toueg [160].

Omission failures refer to instances where a communication channel has failed to perform an action that it was supposed to. An example of this can be seen in almost any large organisation where a breakdown of communication channels leads to projects being delayed, or the project coming in on time but having concentrated its efforts in the wrong area.

Timing failures relate in ever-changing environments where deadlines are missed or work is completed earlier than expected. It can be argued that timing failures are frequently as a result of omission failures, especially in situations where an ever-changing environment exists and activities or specifications are poorly defined.

Byzantine failures represent instances when either messages are not communicated or that the wrong message is sent. Typically, Byzantine failures occur when messages are corrupted or the message that is sent misinforms the recipient.



In any case if any of the three classifications of failure or conflict are discovered corrective action is required. Suitable resolutions will need to be identified by the generic framework and implemented.

#### 5.2.4 Conflict Resolution Strategies

The resolution strategy functionality in the generic framework must exist between the identification and classification of the failure or conflict and the process of understanding the environment within which it exists. K. Barber *et al.* [161] describe the relationship between actions, strategies and strategy decision making in the following diagram:

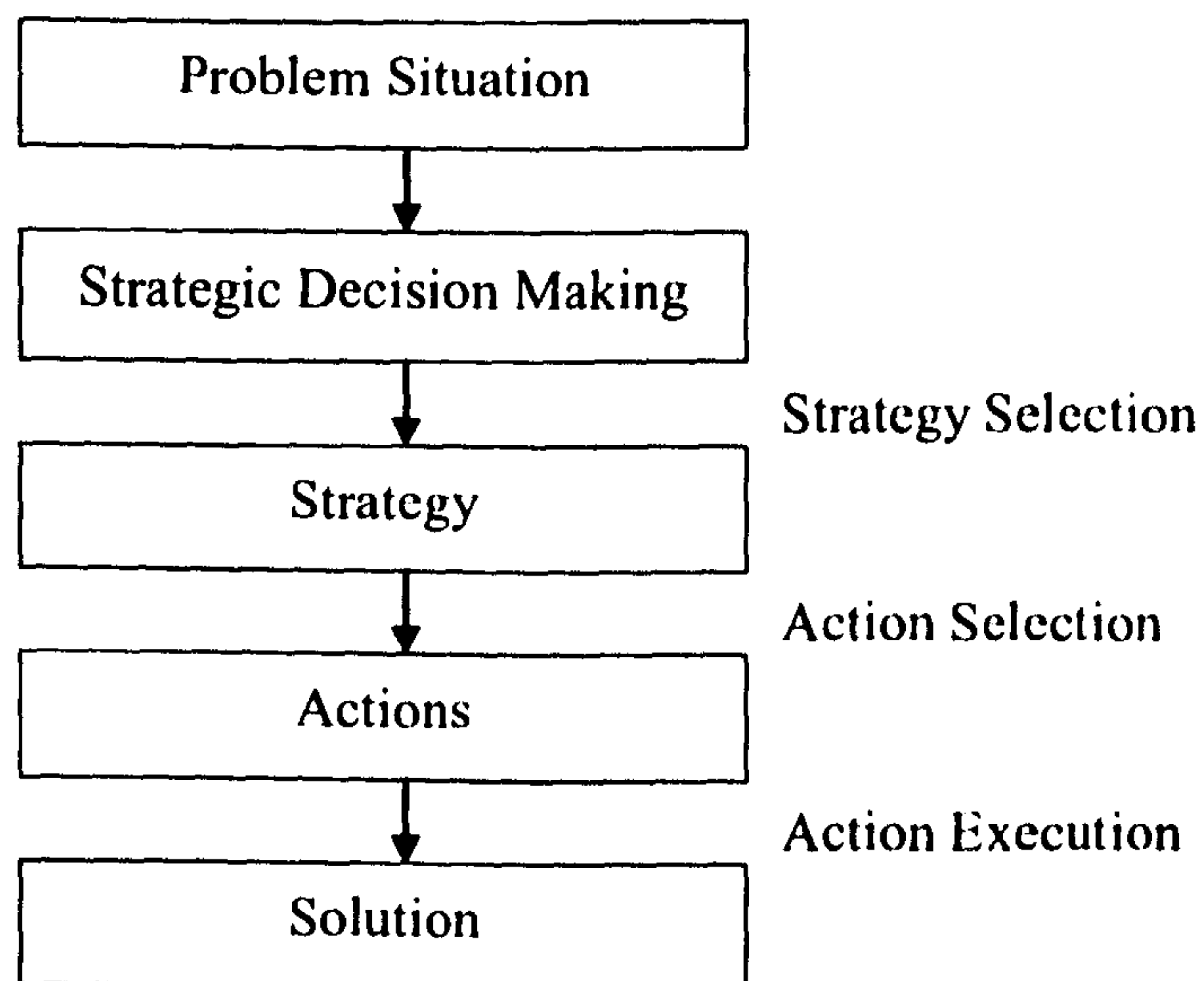


Figure 15: The relationship between actions, strategies and strategy decision making according to K. Barber *et al.*

The appropriate selection of strategies is imperative when attempting to improve the problem situation and avoid any further inconsistencies. K. Barber *et al.* discuss that a rigorous strategy selection process takes into account a number of possibilities. The classification of conflicts such as conflicts that may occur during the plan generation phase or in any of the other phases as indicated in the diagram above. The maintenance of the information system, the environment and the interaction between the two must be taken into account. The maintenance of the information system's preferences cannot be overlooked.

Conflict resolution strategies will be required to be defined and structured as a sequence of plans and events that are considered valid for specific situations that occur within the environment the generic framework is operating in. Essentially the strategies are specified and defined and include details as to where and when they are to be considered appropriate for implementation. The information provided by the detection of a conflict will be checked against current plans and strategies for their appropriateness. This will be conducted to ensure that the chosen plan will be capable of dealing with the problem situation. To this end, it is imperative that the detection and identification of conflicts or failures is performed properly as this will have a significant impact of the selection of an appropriate resolution strategy.

### **5.2.5 Control Rules**

A control rule is usually generated by first comparing two functions or activities within the information system. In this case the usual comparators are the expected beliefs, or current state, and the desired action or activities. Badr [162] defines two types of control rules that are required for establishing any control processes.

External control rules are not attached to any particular strategy but are used for continuous monitoring purposes, such as repair task or execution progress monitoring and coordination. In the event of a conflict during the execution of a control process itself then a conflict is detected and by using a feedback process the system will start another control process.

Internal control rules are embedded within a resolution strategy to monitor the recursion of a strategy's actions or attributes.

### **5.2.6 Reconfiguring the Information System**

An important requirement of the generic framework will be the implementation of a reasoning mechanism for selecting resolution strategies and



control rules, which can be utilised to reconfigure the information system dynamically. As a result the generic framework must have the capacity to coordinate and manage the various activities that will inevitably be taking place to ensure that the information system is working properly within its ever-changing environment.

The ability to negotiate is a prerequisite when the information system is being utilised in a complex environment, in particular one which deals with aspects of culture and communication within large organisations. Badr [163] discusses the requirements for coordination functionality, which should incorporate:

- Shared and accessible by all functions and activities within the generic framework.
- A centralised function which sends and receives requests for strategies.
- The ability to provide a reliable storage capacity for recording all previous requests for strategies.
- The ability to communicate effectively with all other parts of the information system.
- The functionality must be persistent and be in existence at all times.

The ability to apply dynamic communication, management of the problem situation and the potential sharing of resources are important requirements of the generic framework. To this end the ability to reconfigure the information system is assumed to be the most appropriate approach for some conflicts or failures that have been detected.

If a particular conflict situation arises that requires reconfiguring the information system the generic framework will need to be able to generate new or modified activities to implement a resolution. The information systems' desires should be stored thus allowing the various functions and activities to gain an appreciation of what is required when implementing a resolution.

Should more than one resolution strategy be required the generic framework must be capable of providing the necessary support and coordination while the

resolutions are being implemented. There may be occasions whereby the generic framework must choose which resolutions to run at particular times depending on various factors such as available resources and inter-resolution dependencies. In this case there are four approaches proposed by Badr [164] which can be applied when reconfiguring the information system to ensure that the most appropriate resolution is implemented at the right time.

A retry method is sometimes applied if a resolution fails to solve the conflict it is initially better to attempt several retries. If the retries are unsuccessful the generic framework will be required to find another resolution strategy.

Searching must be instigated for an alternative that is considered the best choice if other attempts have failed.

Connecting the information system to a resolution can be achieved if its request in a different part of the environment is understood and meets set criteria.

It must be possible to handle exceptions that are activated as a consequence of any exceptional behaviour that is produced when a control rule executes because of a conflict that has not yet been resolved. Exceptions may be dealt with according to priority to accommodate varying degrees of failure or conflict tolerance.

### **5.3 Chapter Summary**

Recent years have seen a huge step forward in terms of IT systems and their ability to adopt conflict resolution strategies. This is not the case with information systems thinking in a self-adaptive and autonomic sense.

This chapter has discussed the requirements of the generic framework in terms of detecting conflicts and establishing methods to implement appropriate resolutions. By incorporating these requirements the generic framework will be able to accommodate the concepts of self-adaptation and autonomy into the world of information systems thinking.



The main requirements of the generic framework to be developed as covered in this chapter are as follows:

- Conflict detection – this will use a set of control rules against which the behaviour of the information system can be monitored.
- Conflict identification and classification – this function will be activated whenever a conflict or failure has been detected.
- Conflict resolution strategies – the generic framework must be capable of selecting the most appropriate resolution for the problem situation.
- Ability to reconfigure the information system – the capacity of this functionality must be incorporated into the generic framework to ensure that it can apply reasoning when selecting resolutions or control rules.
- Control rules – these must be fully incorporated into the generic framework so that the information system can examine its internal and external process and activities.

The control functionality will be dependent on its control rules and the information gathered by the monitoring activities to detect, identify, classify, interpret and resolve any conflicts or failures. Maintenance of the information system will be achieved by reconfiguring the information system itself within the constraints of its operating environment and the capabilities of its coordination functionality.

## **6. System Design**

### **6.1 Introduction to System Design**

This chapter establishes the target of designing the generic framework. The development of a generic framework will require the successful implementation of the principles and philosophies that form the basis of the two methodologies along with the concepts of self-adaptation and autonomy, which will also need to be incorporated.

Significant amounts of research have been conducted in the self-adaptive and autonomic field looking at the mechanisms involved with controlling problem situations. Osterweil and Clarke [165] established the requirement for a continuous controlling mechanism that adopts the concepts of both feedforward and feedback loops. This is designed to ensure that any software application remains focussed on its original aims. It seems practical that these concepts can be adopted into the information systems thinking world.

This chapter will discuss how the requirements drawn together in Chapter 5 can be incorporated into the generic framework utilising the functionalities of the two chosen methodologies.

### **6.2 The Control Architecture**

This chapter considers the development of the framework's control architecture, or mechanism, which can incorporate self-adaptation and autonomy. Essentially the control architecture will follow three broad strands.

The first strand contains the registration and discovery functionality of the framework. This establishes a monitoring capability and determines when failures or conflicts have been detected.



The second strand contains the management and control of failures or conflicts that have been detected. This requires that accurate information on the problem situation is gathered and contains self-adaptive and autonomic capabilities.

The third strand contains the repair functionality of the framework. This strand contains self-adaptive and autonomic capabilities of the framework and establishes how to repair the detected failures or conflicts.

The whole process starts by implementing fully integrated feedforward and feedback loops into the generic framework. This will form a crucial part of the framework's monitoring and discovery functionality. The importance of these feedforward and feedback loops is that they start a process that must determine the control inputs, check the information system and its operating environment, classify a detected failure or conflict, establish which conflict resolution is most appropriate and ensure that adaptations and information system reconfigurations run smoothly.

To this end, the control architecture, or mechanism, achieves the aims of self-adaptation and autonomy through a continuous cycle of detection, identification, classification and resolution of failures or conflicts. The feedforward and feedback loops will be utilised to continuously monitor the mechanism itself, which ensures that the generic framework has a fault tolerance capacity.

## **6.3 The First Strand of the Generic Framework**

The first strand of the generic framework covers the initial functionality which enables the framework to establish a connection with its environment and implement an appropriate monitoring service.

### **6.3.1 Registration Functionality**

For the registration functionality to operate successfully it must be able to adapt to changes within its environment. It is highly likely that changes to the operating environment will be experienced, especially for larger organisations. As a

consequence the registration functionality must have a certain level of intelligence so as to dynamically alter service level expectations and react to new rules and regulations that are imposed upon it. Having conducted significant into both SSM and VSM it seems clear that the most appropriate functionality of both methodologies is System 4 (Intelligence), which exists within VSM. The diagram below will explain how this functionality fits in with the development of the generic framework.

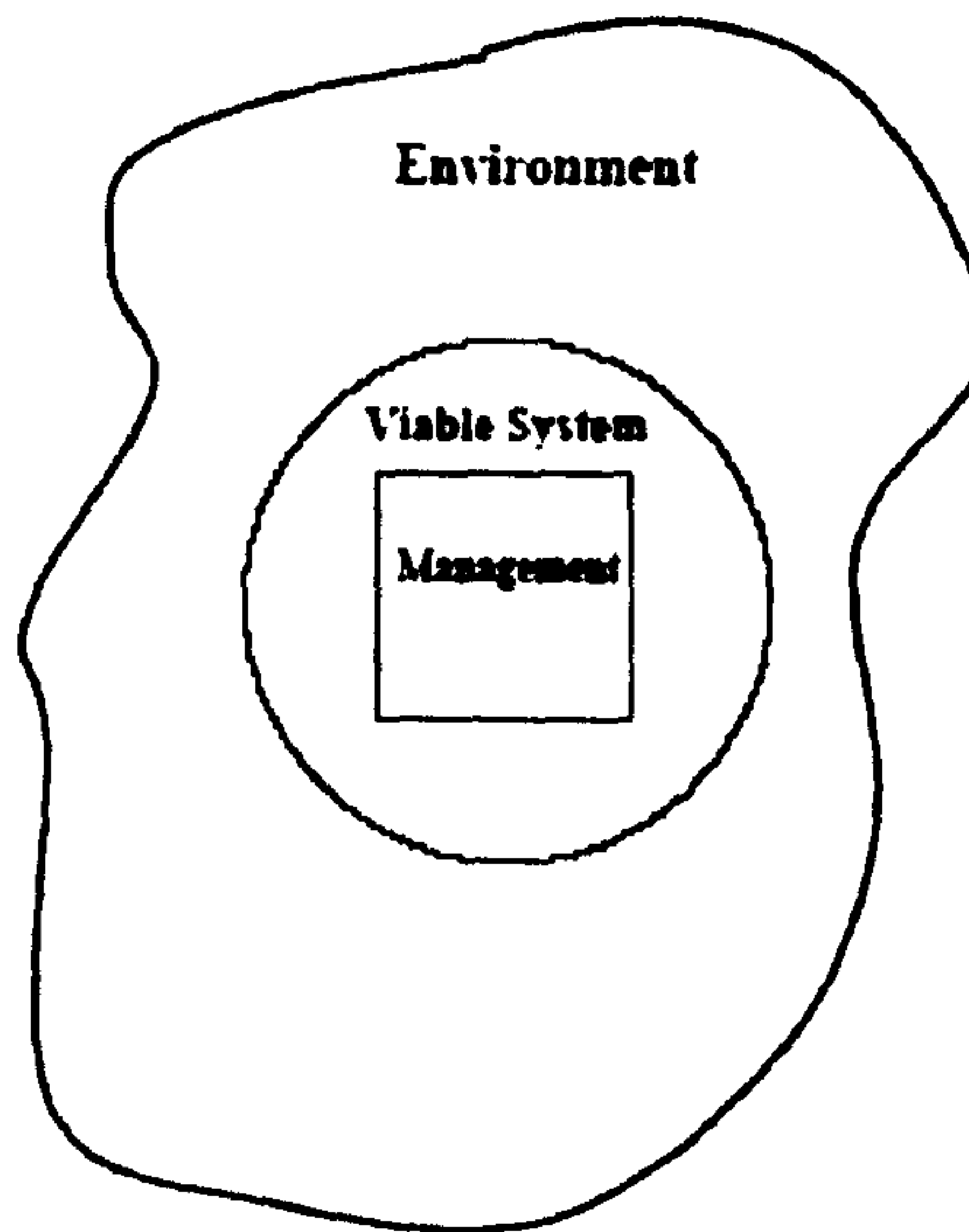


Figure 16: Espejo's [166] representation of a viable system and the components within it.

The registration functionality must reside within the management capacity of the generic framework as this is where service levels and rules and regulations will be determined for the information system. It is System 4 of VSM, which is one of the management functions, that interacts with the environment the information system is working with. This functionality has the responsibility of establishing a connection between the environment and the management capacity of the generic framework.

### 6.3.2 Discovery and Monitoring Functionality

The discovery functionality assumes responsibility for detecting failures and conflicts within the information system and its environment. When failures or conflicts occur it is the internal environment that is the victim as changes in the external environment are generally beyond the information system's control. As a result of this a substantial monitoring function needs to be established that operates



between the internal environment, or viable systems, and the management function. It is considered that SSM's Conceptual Models fulfil the requirements of VSM's Systems 1, 2 and 3\* and can replace them within the generic framework that is being developed.

The Conceptual Models within the Lancaster Model of SSM have the capacity to monitor themselves and feed information both up and down various levels of recursion with the information system. Inherent within the monitoring functionality of Conceptual Models is a set of rules utilised to establish how the information system is performing. These rules are as follows:

- Efficiency – was communication carried out efficiently?
- Efficacy – does this count as improved communication?
- Effectiveness – does this enhance the service offered?

In addition to these three Es the ethics and elegance (the 'other' two Es) must be looked at to ensure that the monitoring process is carried out correctly.

Each Conceptual Model interacts with its environment, which ensures that the information system has a low level of self-adaptation and autonomy and can deal with small scale changes without having to contact the management functionality for a decision – although it will need to contact the management functionality to inform it of a change that has taken place.

The reason SSM's Conceptual Models were chosen over Systems 1, 2 and 3\* from VSM for this function was that the monitoring capacity is more flexible and can be adapted for every viable system that exists within the information system.

## **6.4 The Second Strand of the Generic Framework**

The second strand of the generic framework discussed in this chapter covers the functionality that enables the managing and controlling of failures or conflicts that have been detected.

### **6.4.1 Framework Manager and Control Functionality**

When a failure or conflict has been detected it is essential that information is passed to the manager functionality of the generic framework. From here the manager functionality can determine how to control the current situation. This functionality is essentially a coordinating and cohesion utility which draws together the information provided by the discovery functionality and determines what the next step should be.

Due to the nature of tasks this functionality will be performing it seems most appropriate for VSM's System 3 (Control) to be used at this point. System 3 is the lowest form of management in the VSM and is most in touch with the internal environment in the information system. This is situated here to keep on top of the day to day managing and control of the information system and also looks after any relative rules, regulations and resources that are used by the Conceptual Models.

This functionality takes responsibility logging the information provided and any requests for change that have been made. At this point the manager functionality can determine whether a similar issue has been raised before or whether the current failure or conflict is new. If any further action is required the manager functionality may decide that the next step should be to perform a diagnosis of the problem situation.

### **6.4.2 Diagnosis Functionality**

Once information relating to a failure or conflict has been sent to the manager functionality a decision then needs to be made by the generic framework as to what the next step should be. If the decision has been taken that a diagnosis needs to be performed this functionality will be called into action. This function will be performed by VSM's System 4 (Intelligence).



System 4 is involved with the generic framework's registration functionality and has prior knowledge of service level expectations and frequently interacts with the operating environment within which the information system exists.

System 4 will work with its current knowledge, establish whether there are any relative policies in VSM's System 5 and determine what the diagnosis is having gathered a full appreciation of the current problem situation.

System 4 may also engage itself in various situations and explore different risk scenarios. It must take a proactive view of the protection of intellectual property and the conversion of intellectual capital to intellectual property because the infrastructure is in the process of development and therefore does not yet have requisite variety.

### 6.4.3 Control Rules Functionality

The capacity to deal with unforeseen problems as they arise is key to the success of the generic framework. Whilst the requirements for such a framework have been discussed, along with elements of the framework's functionality that have been mentioned in this chapter, it is important to remember that they need to be incorporated into any developed framework if it is to be implemented successfully.

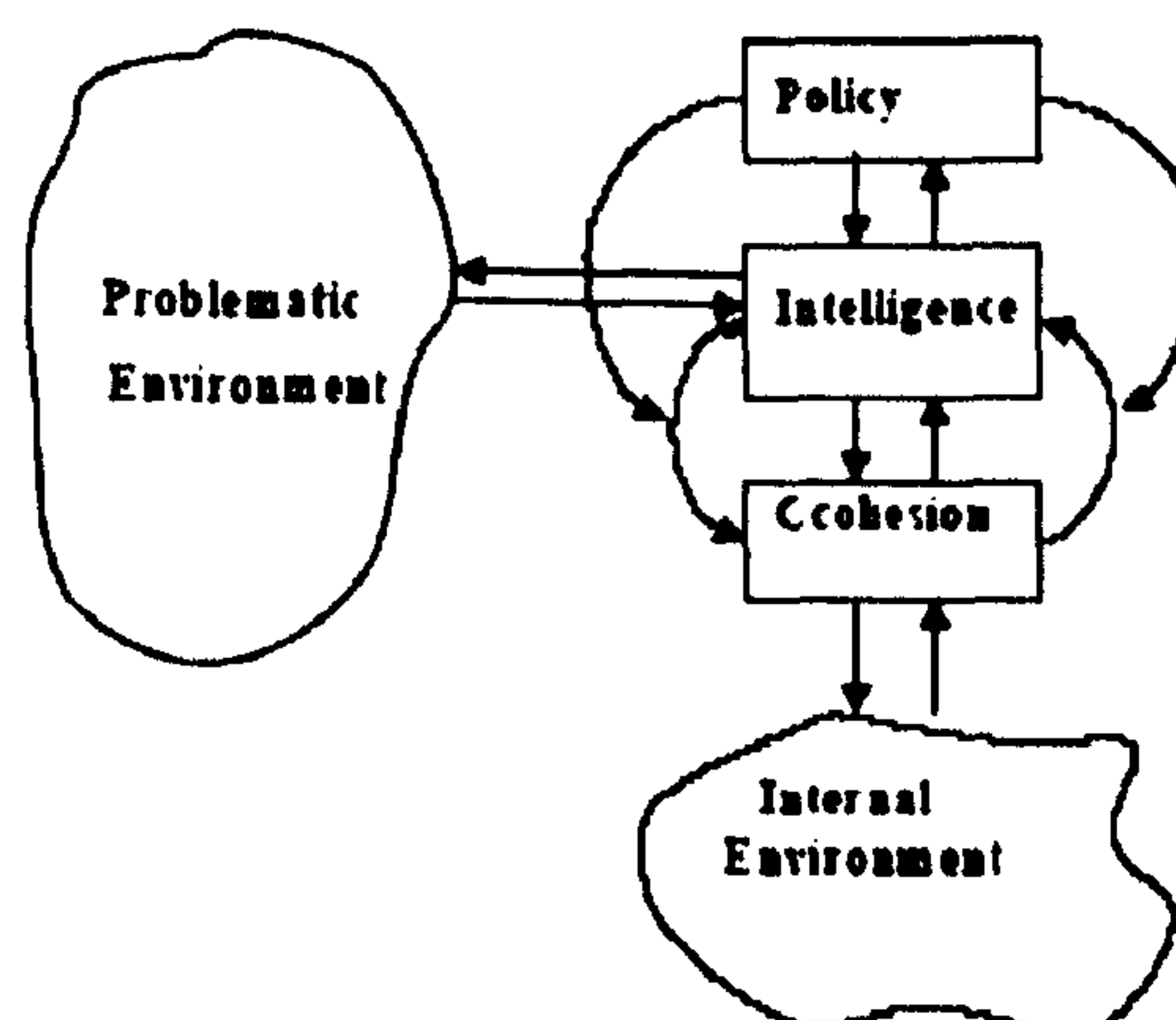


Figure 17: The control rules functionality of VSM as discussed by Espejo [167], which ensures that the generic framework can fully embrace the concepts of self-adaptation and autonomy.

In essence, this leads to the creation of a set of rules which can be utilised in the framework and provide guidance. The development of the set of rules took place to ensure that the generic framework could be used in many different problem situations and was flexible enough to adapt to fast changing environments. Previous information systems thinking, as part of the Liverpool John Moores Enrich initiative [168], has produced a set of rules which are deemed to complement the concepts of self-adaptation and autonomy within the context of this study's generic framework.

The  $R^5$  model suggests the following set of rules need to be applied in order for information systems thinking to fully understand its operating environment, and therefore develop appropriate solutions:

- Resources;
- Rules;
- Responsibilities;
- Regulations; and
- Recommendations

The application of  $R^5$  ensures that the requirements of the generic framework are met and that it is able to fully integrate the functionality discussed in this chapter. In particular, the application of  $R^5$  delivers the ability for the generic framework to incorporate self-adaptation and autonomy through a structured approach that provides guidance for the information system and can also react to problem situations that arise.

Due to the nature of the management structure within VSM the components of  $R^5$  cannot function properly if Systems 3, 4 and 5 work in isolation. The following sections will discuss how Systems 3, 4 and 5 will work together to fully incorporate  $R^5$  and the functionality that it delivers to the generic framework.



### **6.4.3.1 Resources**

System 3 assumes responsibility for the day to day running of the information system and coordinates the low level activities that take place. When a situation arises, which requires additional resources to be allocated, System 3 will communicate with System 4 to determine what is the most appropriate amount of resources to allocate. Before any final decision is taken System 4 will evaluate the current situation within the external environment and perform various risk assessments to ensure that if any other situations arise whilst the resources have been allocated that the information system will be able cope.

### **6.4.3.2 Rules**

In similar fashion to the way resources are controlled Systems 3 and 4 work closely together to ensure that suitable rules are maintained. If any changes in the environment are detected that may require an alteration to the information system's rules then Systems 3 and 4 will communicate with each other to ensure that the necessary changes are made. Notification of the changes will be communicated to System 5 so that the relative policies can be updated. System 5 will always be utilised when rules need checking, updating or when they can assist with decision making.

Equally as important is enforcing the rules should the internal environment no longer adhere to them. System 3 has the responsibility of ensuring compliance with the rules. Should the information system be unable to adhere to the rules a decision will need to be made by System 3 as to how to rectify the situation. If additional resources would be the most suitable resolution then the processes outlined in the previous section, which refers to resources, then Systems 3 and 4 will work together as outlined above. If this problem situation would not be resolved by adopted the resources processes then a decision will need to be made as to what the next step should be and whether implementing a conflict resolution strategy would be appropriate.

### **6.4.3.3 Responsibilities**

During any process that the information system is carrying out, either with the internal or external environments, a mechanism for establishing responsibilities needs to be implemented. An example of this is System 3 takes the responsibility for coordinating and controlling the internal environment on day to day matters.

During the operational lifespan of any information system the monitoring functionality will always take responsibility for communicating any changes in the environment and providing information on any failures or conflicts that have been detected.

It is important that there is a mechanism present for recording who has responsibility for the various functions and activities that are being performed. When a failure or conflict had been detected System 3 will communicate with the body that has responsibility for the relevant functions or activities. As is the case for implementing the resources and rules functionality of R<sup>5</sup> Systems 3 and 4 will communicate with each other and determine what course of action needs to be taken. Decisions will be taken as to whether responsibilities should be reassigned, whether changes in the environment require additional support for the body with current responsibility or whether operational demands mean that the current functions or activities are to be stopped.

As is the case with rules, it is important that System 5 is informed of all those bodies with responsibility and of any changes that may have been made. During the lifespan of the various functions or activities that are being carried out System 5 will constantly be communicated with to ensure that responsibility is still assigned correctly.

### **6.4.3.4 Regulations**

Regulations are utilised in a similar way to rules within this generic framework. Rules are established by the information system and deal with mainly



internal matters. Regulations are generally established from bodies that exist within the external environment with the information system unable to make any alterations to them. This may take the form of laws or policies that are imposed on the information system but must be adhered to.

System 4 naturally takes ownership of this functionality as it is in constant communication with the external environment. Any changes that are made to regulations will be detected by System 4 and communicated to System 5 where the relevant policies and procedures can be updated. System 3 will be informed of the changes that have taken place and what these mean to the internal environment within the information system.

Should any changes to functions or activities being carried out be required as a result of regulation changes then the management functionality must determine what resolutions to implement. In the same way as with the rules functionality of  $R^5$  if the information system is not complying action will be required to ensure future adherence to the regulations.

#### **6.4.3.5 Recommendations**

Recommendations will be made by the management functionality of VSM. Systems 3, 4 and 5 will have shared responsibility for selecting the most appropriate resolutions for any given problem situation.

System 3 takes responsibility for collecting information on problem situations and communicating it with System 4. Once a recommendation has been made System 3 will coordinate and control the implementation of the resolution.

System 4 will assess the external environment and communicate with System 3. Once System 4 has been provided with problem situation by System 3 it will perform an analysis to determine the difference between the desired state and the current state. System 4 will establish with System 5 whether the problem situation has occurred previously and what recommendations were employed then. If the

implemented methods were successful last time then System 4 will recommend to System 3 that they be used again. If the measures were not successful last time, or it is a completely new problem situation then System 4 will determine, with System 5, what rules and regulations are applicable and whether they impose any restrictions on a possible resolution. Taking due cognisance of the problem situation System 4 will then make a new recommendation and provide System 3 of the details of what measures are to be implemented. System 5 will then be updated with details of the problem situation, the resolution and its level of success.

System 5 is provided with details of the full problem situation and has a full record of all the relevant rules and regulations. In many cases the rules and regulations will determine what resolution will be implemented, with the details being passed to System 4. If the rules and regulations allow some flexibility System 5 will work with System 4 to assess whether this problem situation exists elsewhere in the environment and what resolutions were implemented. If these resolutions were successful then Systems 4 and 5 will make the decision to adopt these measures and fully document the decision that was taken. If the resolutions were not successful elsewhere the Systems 4 and 5 will decide what resolution will be the most appropriate, document the decision that was taken, and provide System 3 with the necessary details.

The successful implementation of any resolution will require effective monitoring and controlling functionality to ascertain how the implementation is working out and whether the current resolution has led to new problem situations arising in other parts of the information system.

## **6.5 The Third Strand of the Generic Framework**

The third strand of the generic framework covers the repair functionality of the framework. This strand contains self-adaptive and autonomic capabilities of the framework and establishes how to repair the detected failures or conflicts.



### 6.5.1 Repair Functionality

The repair functionality takes responsibility for implementing changes that resolve problem situations. Through utilising the concepts of self-adaptation and autonomy, as previously discussed, the ability to self-repair can be incorporated into the generic framework. As the generic framework is being constructed to include all the functionality discussed in this chapter it can be seen how the three strands of functionality are beginning to come together.

With regards to the repair functionality there are three key tools available, which require cooperation with the other two strands built in to the generic framework. S. Krishana *et al.* [169] discuss three key tools for dealing with detected failures and conflicts and how to implement the most appropriate resolutions.

Notification functionality enables interested parties, in this case either the internal environment or the controlling functionality in System 3 of VSM, to register their interest with particular events and be provided with any relevant updates. This functionality provides fast and effective communication to all relevant activities within the information system.

A repair capacity enables the functionality that implements selected resolutions for various problem situations. The ability to self-repair can be incorporated into the generic framework at this point whereby it can select resolutions dynamically through utilising the management functionality of the generic framework. This allows the information system to adapt to changes within its environment when utilised within certain established boundaries.

Exception handling capability working in a similar fashion to the repair functionality, exception handling can initiate certain functions or activities to resolve particular failures or conflicts that are known to the information system. The management functionality of the generic framework will establish a set of priorities through implementing the  $R^5$  approach, which leads to a fault-tolerance within the information system.

### 6.5.2 Self-Adaptive Functionality

The management of self-adaptive functionality is crucial to the success of the generic framework. The management functionality of VSM has the capacity to adapt to changes but for the generic framework to achieve its full potential it must incorporate the concepts of self-adaptation and autonomy.

This functionality can be fully adopted by the generic framework through an effective adaptation strategy. M. Kokar *et al.* [170] state that an effective adaptation strategy is required where the parameters are managed and controlled by a predefined limitation of the information system parameters and a range of the variation of possible adaptation. It is also key to prevent more changes arising by reducing the number of functions or activities required for information system planning or developing strategies.

There is an approach that can be adopted which creates the adaptation process, as defined by D. Pautler *et al.* [171], which requires the following two conditions to be in place:

- Pre-condition – the required activity type, representing the limitation or other constraints on parameters that are required for the function to perform its adaptation.
- Post-condition – the effective monitoring of the generated activity types implemented.

Utilising this functionality the generic framework will ensure that the information system can send and receive messages relating to the success or otherwise of the implemented adaptive activities. This requires powerful feedforward and feedback loops, which exist within the generic framework, and the ability to monitor and communicate with the external environment simultaneously.

To this end, the functionality of VSM's System 4 is deemed the most appropriate to utilise as it can fully incorporate the self-adaptive functionality and can constantly determine which parameters are changeable and which must not be altered.



The feedforward and feedback loops play a significant role in the process as they send results of the adaptation back to the monitoring and control functionality to determine the success of the selected repair activities.

### **6.5.3 Repair Strategy Functionality**

Repair strategies are crucial to the generic framework given that it is utilising the concepts of self-adaptation and autonomy. Self-adaptive functionality, as discussed in the previous section, requires definitions to be in place to help it determine when, where and how the repair work, and self-adaptation, is required. Even dynamic information systems require predefined resolutions to be available for selection. This is why repair strategies are utilised as they help reduce the amount of options available for selection and, most importantly, reduce complexity whilst still allowing the framework to operate in an adaptive and generic way.

Repair strategies must take due cognisance of the activities present within the information system and its environment. To ensure that this is incorporated into the generic framework the concepts of the BDI Model (Beliefs, Desires and Intentions, as discussed in Chapter 3) are used. In this case the resolution strategies are used to represent the effect of various alternative solutions on problem situations, as discussed by S. Rao and P. Georgeff [172].

Beliefs are represented by two structures. Firstly, a model of the external environment and secondly the internal environment that exist within the information system. Desires represent a set of actions and desired aims which need to be achieved at a specific time. Intentions are determined by a process of deliberation, which translates desires with respect to the current beliefs about both the environment and the current weltanschauung.

In effect the generic framework will be using repair strategies as a set of executable activities or functions. To fully encompass this functionality System 5 of the VSM will be used as it contains all the necessary functions and activities required for repair strategies. To this end, each repair strategy will be examined by pre-

conditions following S. Rao and P. Georgeff's [173] approach. This extends to being able to examine parameters, properties and rules of each activity and the ability to determine the applicable strategy to resolve the failure or conflict. Strategies are formed out of lower level activities based on the BDI Model, as described above. The information system control rules determines whether or not this particular strategy matches the information system Desires and selects the appropriate Intentions when the information system's Desires are compared with its current Beliefs.

Selecting one valid strategy in a problem situation where several may be applicable should be decided according to policies as defined by System 5 of VSM.

Reconfiguration of the whole information system after selecting the appropriate solution is the essential phase. The information system repair strategy should notify the reconfiguration processes and activities of the required reconfiguration needed to establish and provide an accurate representation of the applicable activity configuration.

#### **6.5.4 Reconfiguration Functionality**

It is essential that the generic framework leads the information system to establish whether the selected resolutions strategies have been successful. The ability to reconfigure an information system is important as it allows it to cope with changes in its environment. As a consequence, it stands to reason that a self-adaptive and autonomic generic framework should be able to reconfigure itself. This functionality extends to carrying out reconfiguration of the information system whilst it is in operation.

The information system reconfiguration applies the required reconfiguration via the repair strategy, as discussed in the previous section of this chapter. The autonomic control of the reconfiguration process requires the resolution to be closely monitored and changed if it, or any part of it, is failing to achieve its aims.



To ensure this functionality can be accommodated within the generic framework it is essential that reconfiguration ties in very closely with the information system's repair strategies. Therefore, System 5 of VSM will be utilised, as it is for the generic framework's repair strategy functionality, to carry out this duty. This will also ensure that previous resolutions have been attempted but have failed the reconfiguration functionality will be made aware of this and another resolution will be implemented.

## **6.6 Chapter Summary**

This chapter has drawn together the requirements collected and used them to design the generic framework and functionality that resides within it. The concepts of self-adaptation and autonomy are crucial to the generic framework, which is why they are included in the design.

Such an approach allows the generic framework to utilise existing information systems thinking whilst embracing new functionality facilitated by linking the concepts and philosophies of SSM and VSM.

The following chapter provides a detailed account of the development of the generic framework based upon the design process that has been discussed.

## **7. System Development**

### **7.1 Introduction**

Taking into consideration the requirements discussed in this thesis and the functionality described in the previous chapter the process of constructing the generic framework can begin. An understanding of the 'whats' and the 'hows' of problem situations are fundamental to the success, or otherwise, of the information system. The 'whats' of the problem situation must be understood and defined first as decisions made with regards to the 'hows' cannot be made until the first stage is completed. As a consequence of this, and due to the inherent philosophy within SSM, using the Lancaster Model as the basis for the generic framework seems to be the most appropriate place to begin. This will allow the generic framework to ensure that an accurate understanding of the problem situation within the information system can be properly defined before attempting to implement resolutions.

### **7.2 Building the Generic Framework**

The following sections of this chapter discuss the development of the generic framework incorporating the requirements drawn together in this thesis.

#### **7.2.1 Utilising SSM's Lancaster Model**

Not only does the Lancaster Model from SSM define the problem situation it also allows the application of other systems thinking in Step 4b, as shown below.



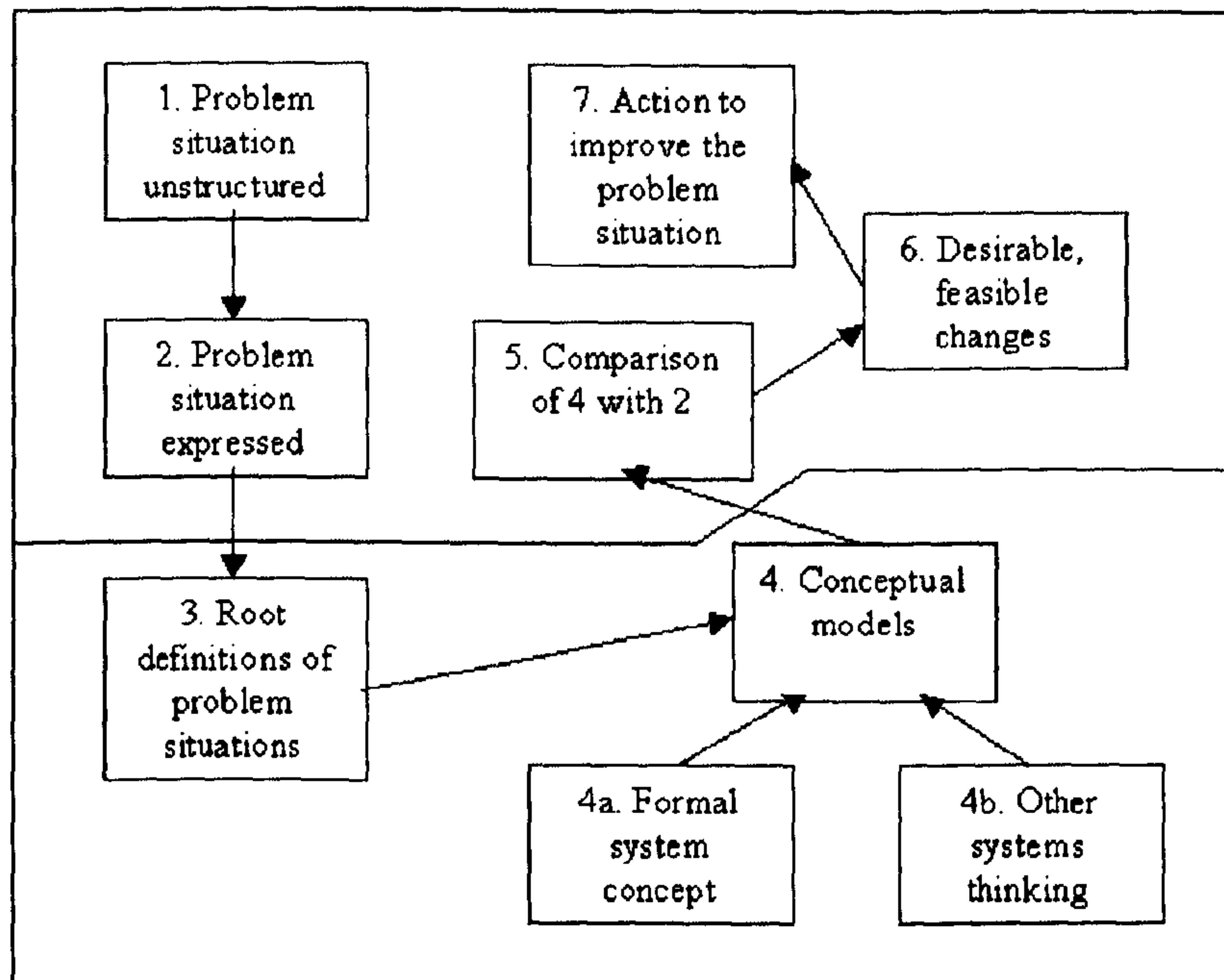


Figure 18: Adaptation of the Lancaster Model produced by Checkland, as discussed in Chapter 2 [174].

The introduction of VSM into the Lancaster Model has been discussed previously in the world of information systems thinking by Kinloch and Francis [175]. It is considered appropriate for the Lancaster Model to be used as the base for which to develop the generic framework as it will accommodate other systems thinking whilst also ensuring that the fundamental philosophies that underpin SSM are still contained within anything that is developed.

For the purposes of creating the generic framework, and because of SSM's well established approach, it is desirable to ensure that Steps 1, 2 and 3 of the Lancaster Model are kept in their entirety. The first three steps of the Lancaster Model are proven to be successful in defining a structured approach to determining the 'whats' of a problem situation.

Step 4 of the Lancaster Model is where Conceptual Models for information systems are defined, with the opportunity here to incorporate other systems thinking. This is where the Viable System Model is to be incorporated into the generic framework.

### 7.2.2 Incorporating the Viable Systems Model

As discussed in the previous section of this chapter Step 4b of the Lancaster Model can accommodate the functionality from other systems thinking. It is at this point where VSM will be incorporated into the generic framework. The philosophies that underpin the VSM will establish the 'hows' that form part of the problem situation.

It is proposed at this point that the Lancaster Model adopts some changes to ensure that both SSM and VSM are fully integrated with each other and that a new Step 4 is developed for the purposes of the generic framework that is being constructed here.

#### 7.2.2.1 Overlapping Functionality

The concepts that underpin SSM's Conceptual Models fall within the boundaries defined by Systems 1, 2 and 3\* of VSM. The definition of a Conceptual Model by the Operational Research Society [176] is as follows:

*"...Conceptual Models are a way of expressing logically derived ideas about what systems should be in place in an information system, according to the root definitions. The models will be used in later stages as bases for discussing what is really happening, and for finding ways to deal with the problem situation."*

There is a significant overlap in approaches between Conceptual Models and Systems 1, 2 and 3\* of VSM. In VSM Systems 1, 2 and 3\* take responsibility for defining what systems and activities are in place within information systems, how they are performing and feeding the information to the control functionality. The diagram below shows basic approach adopted by VSM.



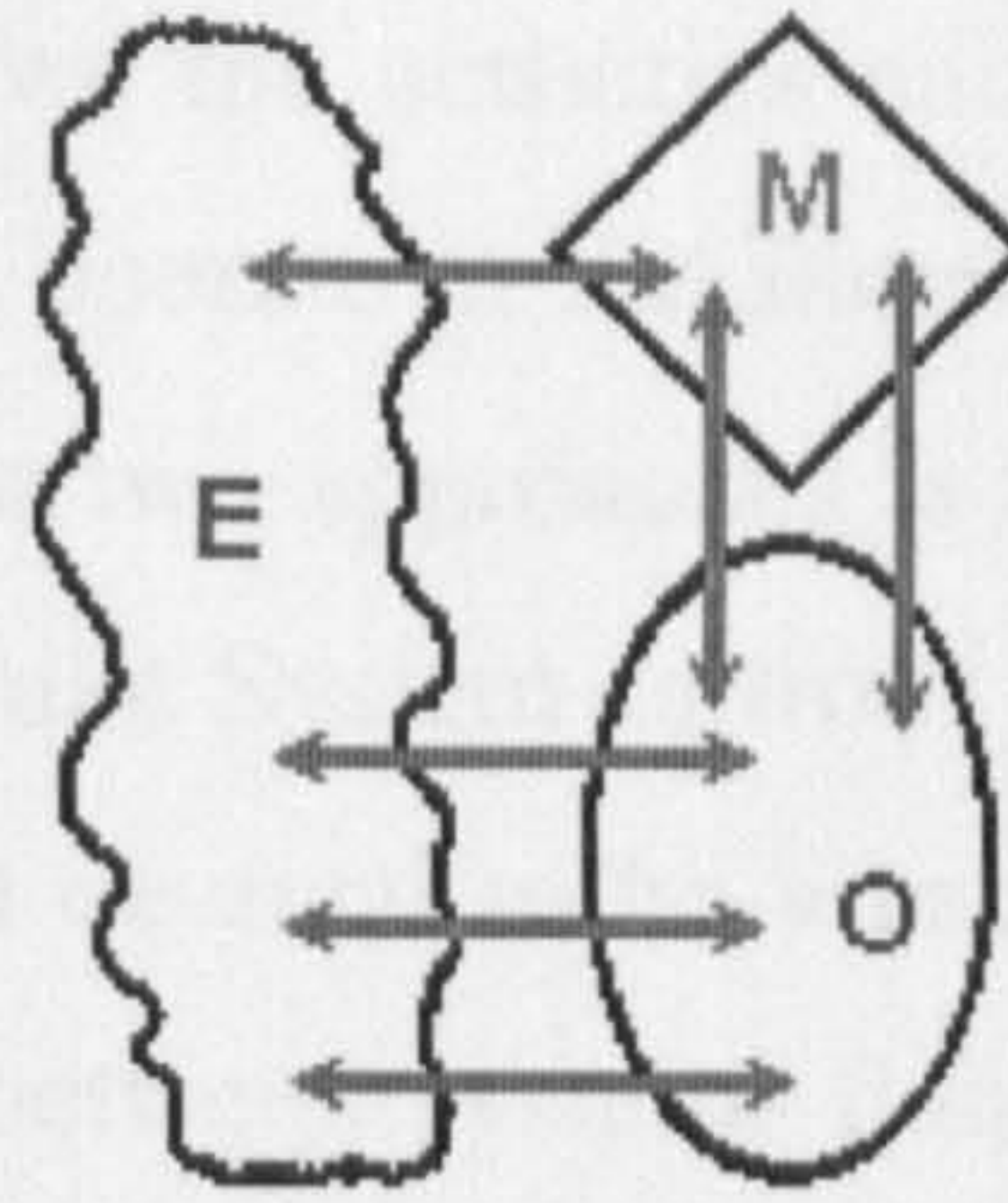


Figure 19: Meta level depiction of VSM [177].

In this instance E represents the external environment, O represents the operations (or internal environment) and M represents the metasystem, which is the management functionality of VSM. The arrows indicate the various ways that the three parts of the model interact with each other.

In many ways the approach utilised by Conceptual Models follows the same lines as the internal environment of VSM, which by its own definition is Systems 1, 2 and 3\*. The diagram below shows the Conceptual Model and its component parts.

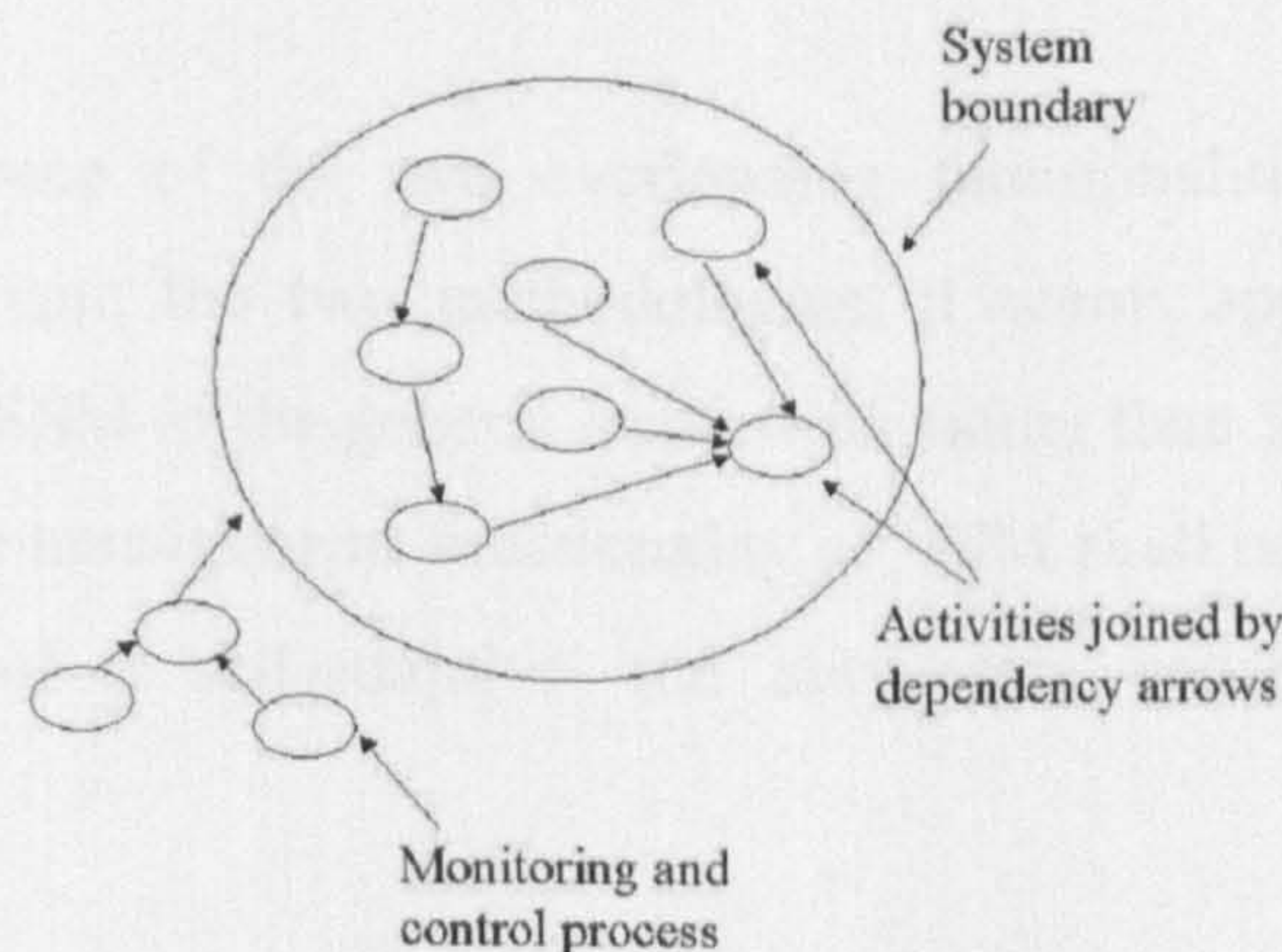


Figure 20: Conceptual Model of SSM and its component parts by Brooks [178].

The system boundary, as defined by Steps 2 and 3 of the Lancaster Model, ensures that the Conceptual Model is relevant to the information system's aims and objectives and exists in its entirety within the environment, which is effectively a viable system. Conceptual Models allow several levels of recursion, which is exactly the same for viable systems in VSM.



The diagram above shows the activities and functions that are incorporated into Conceptual Models, which operate in the same way that System 1s in VSM do. The only difference between the two approaches is that Conceptual Models are used to define the 'ideal' situation whilst System 1s from VSM describe what is happening in reality. However, this is not deemed to be a problem as Step 5 of the Lancaster Model conducts a comparison between systems thinking and the real world problem situation.

The monitoring, evaluation and control of information systems can, at a local level, be carried out by the Conceptual Model's monitoring functionality. This works in the same way that Systems 2 and 3\* of VSM do. The 3Es of Conceptual Model's monitoring functionality ensure that low level decisions can be made against defined monitoring criteria, whilst the opportunity is available to communicate with higher level functionality should a decision need to be made which might affect other information systems in the internal environment too. If this is the case the monitoring functionality of Conceptual Models relays the gathered problem situation information for conflict interpretation and identification.

As a consequence of the two overlapping functionalities, and given the philosophies that underpin the two methodologies, it seems appropriate to utilise Conceptual Models of SSM in the generic framework rather than Systems 1, 2 and 3\* of VSM. However, the management functionality of VSM shall be fully integrated to ensure the concepts of a self-adaptive and autonomic generic framework are incorporated.

#### **7.2.2.2 Incorporating Decision Making Functionality**

Whilst it has been shown in the previous section of this chapter that there is some low level decision making capacity it is a prerequisite that the generic framework has fully integrated management and coordination functionality. To achieve this aim, and also the aim of incorporating self-adaptive and autonomic functionality, it is proposed that Systems 3, 4 and 5 of VSM are utilised in the generic framework.



This is the first step in the development of the generic framework whereby there is a transition from establishing the 'whats' to the 'hows' of problem situations. This chapter has discussed how the management functionality of VSM integrates with both the internal and external environments within which the relevant information system exists.

The monitoring functionality of the Conceptual Models will provide accurate and up to date information for the management functionality to perform in relation to selecting any necessary control rules or repair strategies. The diagram below shows how Conceptual Models and the management functionality of VSM will work together to take the first step towards combining the 'whats' and the 'hows'.

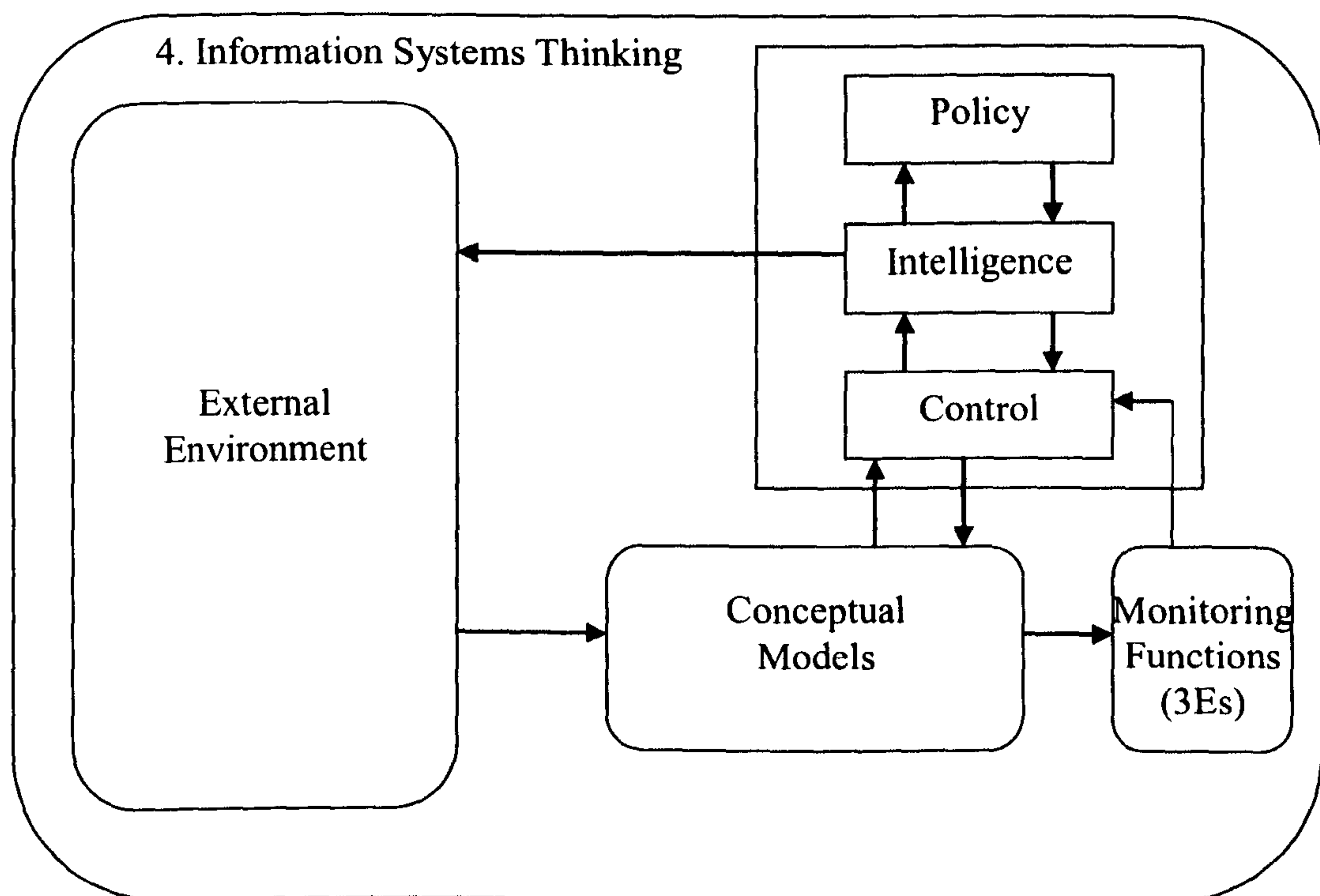


Figure 21: Diagram to show how Conceptual Models and the management functionality of VSM will work together in Step 4 of the Lancaster Model.

### 7.2.2.3 Integrating Self-Adaptive and Autonomic Concepts into the Generic Framework

To ensure the generic framework is fully self-adaptive and autonomic it needs to use  $R^5$  in Step 5 of the Lancaster Model so as to conduct appropriate analysis between systems thinking and the real world situation. This will ensure the third

strand of the control architecture of the generic framework is included which aims to ensure that an appropriate repair strategy is implemented.

In essence, Systems 3, 4 and 5 will perform the duties of  $R^5$ , however, it should be noted that in any self-adaptive or autonomic information system the ability to override the decision should be present. The intention of the generic framework is for it to develop and improve information systems, and the problem situations that go with it, in such a way so that it is able to lead itself to make decisions based on information it gathers. However, these decisions can always be overridden should it be deemed manually that another resolution is more suitable for the failure or conflict currently being experienced.

It could be said that Steps 5, 6 and 7 of the Lancaster Model are effectively carried out by management functionality of VSM in this generic framework. However, to ensure that there is always the option to override decisions that have been made Steps 5, 6 and 7 of the Lancaster Model will be kept as part of the generic framework.

The findings from Step 5 of the Lancaster Model, upon completion of the  $R^5$  analysis, will lead to resolutions being established and put forward for selection. These resolutions will then be selected and carried out within the information system and constantly monitored to see what effect they have on the problem situation.

### **7.3 Introducing the Complete Generic Framework**

Having fully integrated the capacity to make decisions and incorporated the concepts from self-adaptation and autonomy the generic framework the process of development can be considered done for the time being. This stage will, of course, be reviewed once the generic framework has been tested, which is covered in the following chapter, and any necessary alterations to the generic framework will be made.



The requirements and functionality incorporated into the generic framework will provide a new approach to information systems thinking, which in turn should see information systems operate better within their working environments. The newly developed Step 4 of the Lancaster Model will enable the generic framework to reduce complexity whilst communicating effectively with both the information system's internal and external environment.

Introducing the concepts put forward in  $R^5$  will enable the generic framework to lead the information system to determine what resolutions should be implemented based on the information provided and experience gained from previous failures or conflicts.

It should be noted that the Lancaster Model is an iterative process whereby once Step 7 has been completed it is possible to return to Step 1 and revisit the problem situation to establish how successful or otherwise the implemented resolution has been. Successful application of the generic framework requires that constant monitoring is performed, within predetermined boundaries, which is why the use of such comprehensive monitoring and control functions have been included.

Taking cognisance of the requirements and functionality discussed the generic framework will take the form as shown in the diagram below.

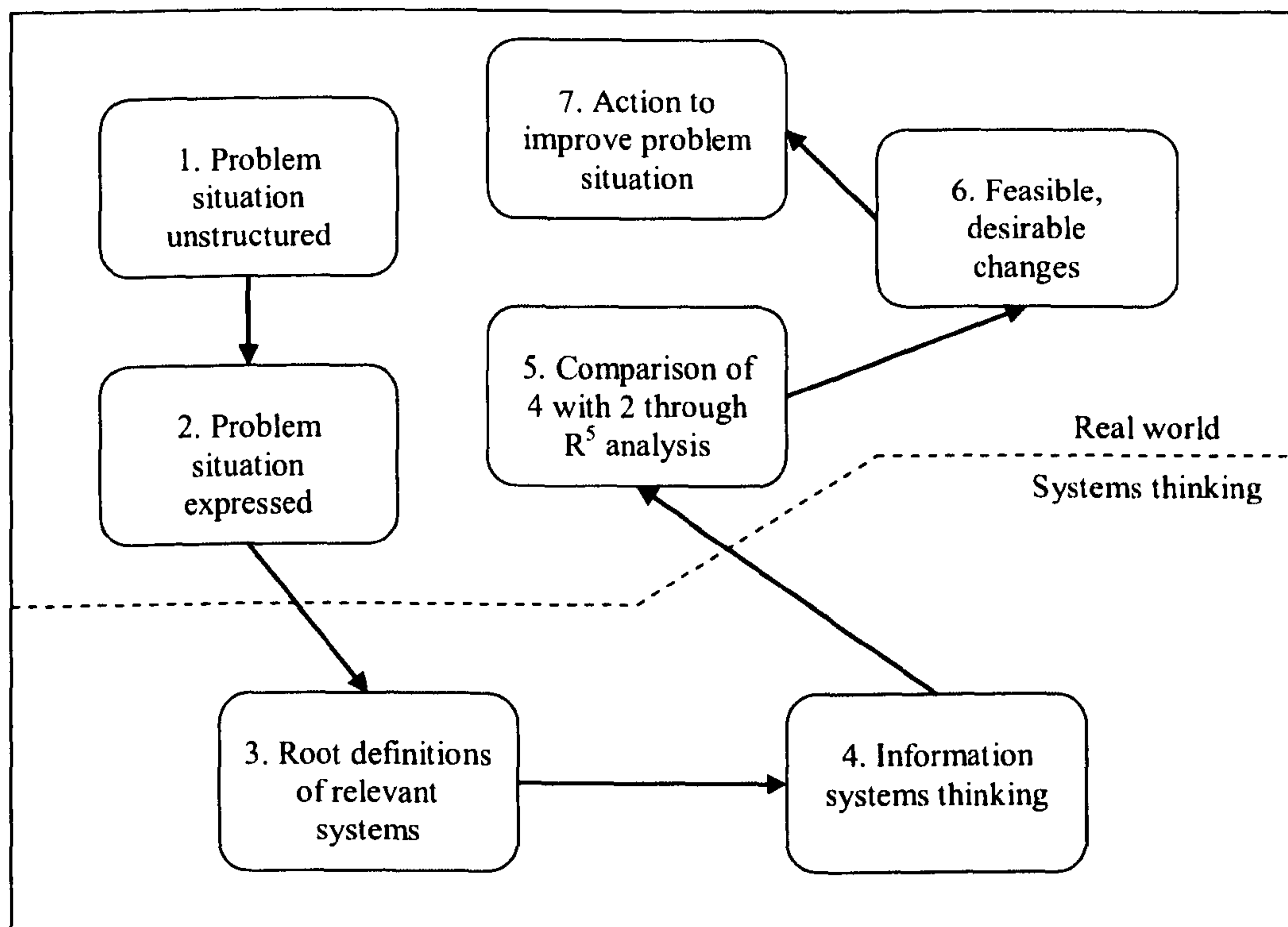


Figure 22: Diagram showing how the new generic framework fits within the Lancaster Model of SSM.

The intention, when developing the generic framework, was to ensure that the ‘look and feel’ of the Lancaster Model was kept intact, as it has been a successfully adopted approach to information systems thinking for several decades. Whilst the generic framework may look similar to the Lancaster Model it should be noted that the functionalities, as discussed in Chapters 4 and 6, are fully incorporated into Steps 4 and 5 whilst ensuring that the philosophies that underpin SSM and VSM are not compromised.

## 7.4 Chapter Summary

The development of the generic framework incorporates the requirements discussed in Chapter 5 and the functionality described in this chapter. The ability to effectively deal with failures or conflicts within the information system, whilst incorporating self-adaptive and autonomic functionality is fundamental to the generic framework.



The main strength of the generic framework is that it is flexible enough to be utilised in many different situations and is able to manage, control and coordinate resolutions to failures or conflicts that arise. Conventional information systems thinking methodologies and approaches can be combined to ensure effective development, management and control of information systems, as shown in this chapter.

The generic framework introduces mechanisms and processes which are aimed at reducing complexity whilst allowing information systems the ability to manage and control themselves within the limitations of defined boundaries and policies. The application of  $R^5$  to the generic framework establishes a set of rules, which lead the information through the various steps needed to ensure that appropriate resolution strategies are implemented.

## **8. Experimentation of the Generic Framework**

### **8.1 Introduction to the Experimentation of the Generic Framework**

This chapter aims to provide details of the experimentation with the generic framework. To assess the effectiveness of the newly developed generic framework it is essential to conduct testing through implementing it in appropriate environments. In such circumstances it is necessary to select case studies that are experiencing difficulties and applying a new approach to information systems thinking may prove to be a helpful solution.

To test the generic framework, and its adaptability to differing scenarios, it is deemed appropriate to implement it in a large organisation environment. For the purposes of experimenting with the generic framework Merseyside Police has been selected as the organisation to work with. This chapter covers the application of the generic framework to Merseyside Police. However, this represents several case studies due to the scale and complexity of the issues arising that were challenged by the generic framework within Merseyside Police. As will be discussed in the following sections of this chapter the two main case studies conducted were based around tackling the information starvation and cultural issues within the organisation. This provided an appropriate testing environment for the generic framework as it incorporated both a hard (information starvation) and soft (organisational culture) problem for the generic framework to work with.

### **8.2 Case Study: Merseyside Police**

This chapter aims at establishing the main problem situation being experienced within Merseyside Police. Whilst establishing the problem situation this chapter also introduces the organisation and sets out how Merseyside Police had arrived at the position it found itself in.



Evidently the problems being experienced by Merseyside Police are indicative of large organisations, and specifically of large public sector organisations, which have stringent legislative regulations imposed upon them. Due to the size and complexity of activities carried out in the day-to-day running of such an organisation Merseyside Police was chosen as the case study. Such an organisation would present an excellent opportunity to test the generic framework and any new information system that has been developed. The size and scale of Merseyside Police would require that the generic framework's functionality would be fully tested and that any deficiencies would be highlighted.

## **8.2 Introducing Merseyside Police**

In 2004 Merseyside Police set itself the vision of becoming the best police service in the UK. Chief Constable Bernard Hogan-Howe stated that a Blueprint for Service Improvement [179] would be implemented to help the organisation achieve this ambitious aim. The blueprint contained ten areas where the Force needed to develop itself in order to achieve the aim of being the best police service in the UK. The ten areas are as follows:

- Blitzing antisocial behaviour (ASB);
- Improving service delivery through calls and response;
- Using technology to fight crime;
- Tackling gang related crime;
- Improving crime reporting;
- Improving criminal justice processes;
- Professionalising the workforce through academy training;
- Extending the police family (including Police Community Support Officers, Special Constables, volunteers and Cadets);
- Road safety; and
- Improving citizen focus.



To further develop the ten areas highlighted above a concept of total policing was introduced. At its core is a commitment to utilise the total resources of the Force and direct them to three strategic priorities. These are:

- Total war on crime;
- Total care for victims; and
- Total professionalism.

The Force has a budget establishment for just over 7,000 police officers and police staff, which are spread across the organisation in six Basic Command Units (known as BCUs).

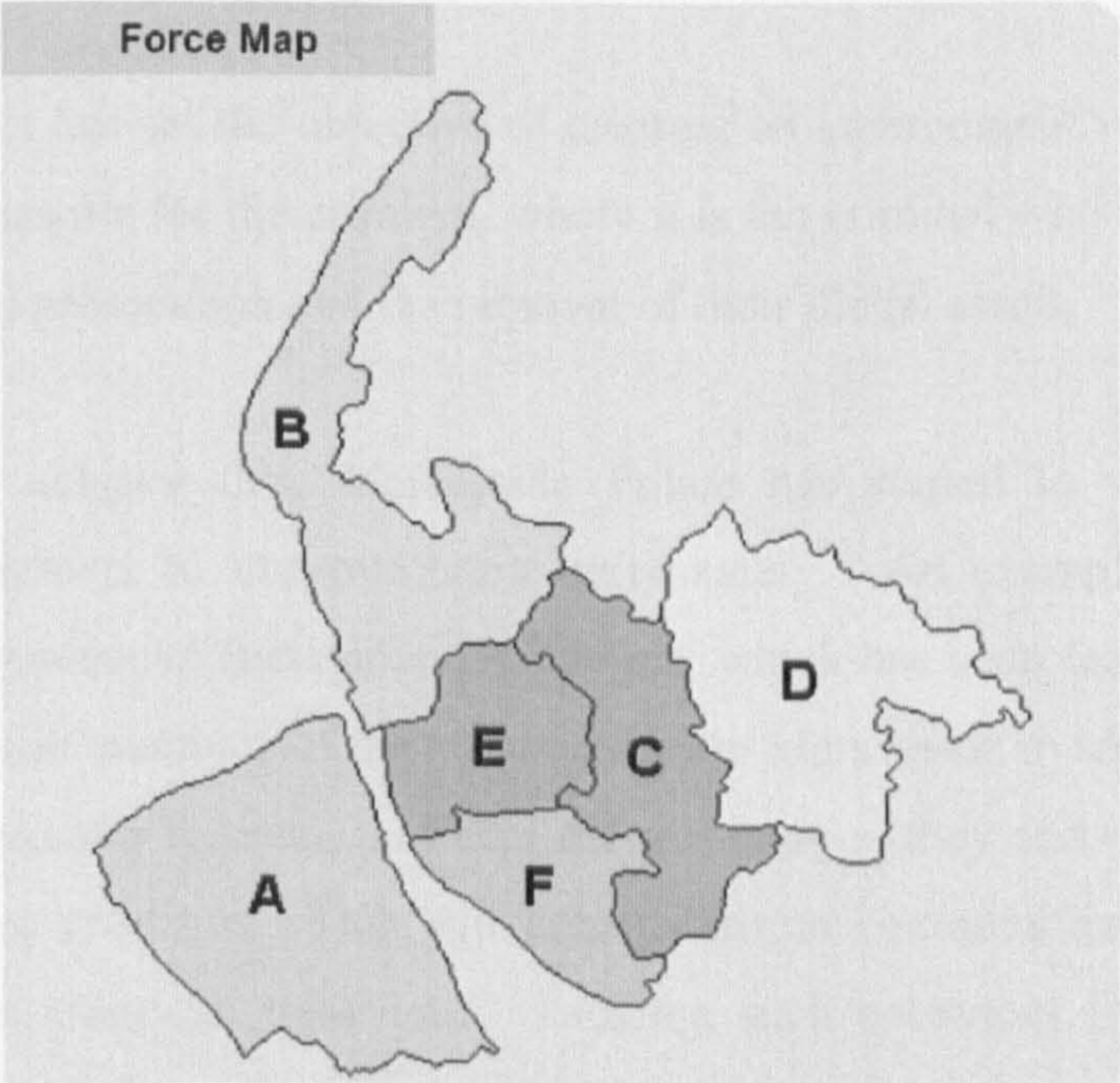


Figure 23: Breakdown of BCUs across Merseyside Police [180].

Basic Command Unit	Area
A	Wirral
B	Sefton
C	Knowlsey
D	St Helens
E	Liverpool North
F	Liverpool South

Figure 24: Division structure of Merseyside Police [181].



### **8.2.1 Total War on Crime**

Criminal behaviour, whether from organised criminal gangs or people involved in low level antisocial behaviour, undermines local communities, creating a climate of fear and mistrust according to the Blueprint for Service Improvement [182]. Under the new approaches underlined in the blueprint for service improvement Merseyside Police is committed to a total war on crime, where opportunities for crime in all its forms are reduced, the law is robustly enforced and offenders are arrested and successfully prosecuted. The Force wants to minimise the fear of crime within Merseyside's communities, enhance feelings of reassurance and safety, and increase the public's confidence and satisfaction in their police service.

The Force has set the objective of creating an environment where there is no hiding place or respite for the criminal, where it is the criminal who lives in fear – of arrest, successful prosecution and the removal of their illegal assets.

To help achieve this, Merseyside Police has started to work with local agencies and partners to improve community safety. An example of this is the creation of the Antisocial Behaviour Task Force, which has been formed. This Task Force engages with members of the communities in Merseyside to identify issues that are a cause of concern to them, and uses the information they and others provide to establish policing priorities. This will help to ensure communities do not have to suffer long term antisocial behaviour. Tackling such behaviour is fundamental to creating safe, sustainable and prosperous communities.

During the period of this approach Merseyside Police's aim has been to start a total war on crime and to target, disrupt and bring to justice those who are involved in criminal activity. Legislation is being used to strip criminals of their assets and illegal profits and deny them free use of the roads. Heavy investments have been made into forensic science to improve the quality of evidence, and in current and emerging technology to ensure staff can access the information they need both in the office and out on patrol.

### **8.2.2 Total Care for Victims**

The total care for victims policy places the needs of the victim, witness and law abiding citizen at the heart of the service provided. In the Blueprint for Service Improvement it recognises there is no 'one size fits all' and that victims require a service tailored to their individual needs, albeit built upon minimum service standards.

Merseyside Police has committed itself to doing everything that it reasonably can to meet the needs of victims, witnesses and law abiding citizens. Total victim care aims to encourage this by building public confidence and trust in Merseyside Police. To achieve this Merseyside Police will continue to work on those things that contribute to public confidence including:

- Engaging with communities;
- Being available at the right times in the right ways;
- Ensuring neighbourhoods are policed well;
- Providing a good service; and
- Being fair in our dealings with people.

From research the organisation knows that as a priority it needs to:

- Improve the service Merseyside Police provides to victims of road traffic collisions;
- Address differences in the satisfaction levels between black and racial minority groups and white users with the service they receive;
- Provide more follow up information for all crime victims;
- Improve call handling and response times; and
- Improve the way in which antisocial behaviour is dealt with.

Merseyside Police is concerned with implementing mechanisms from the initial call for service through investigation and beyond. Improvement against this commitment underpins the delivery of responsive and accessible services at



neighbourhood level across the constabulary. Increasing the levels of reported satisfaction is one of the key indicators for the provision of a citizen focused police service according to the blueprint for service improvement.

In 2004 Merseyside Police, as stated in the Blueprint for Service Improvement, adopted the national reassurance model as an integral part of neighbourhood policing. This ensures that the organisation engages and works with local people and communities to establish policing needs, address their priorities and provide feedback.

### **8.2.3 Total Professionalism**

Total Professionalism in Merseyside Police is aimed at developing an efficient, effective and motivated workforce, which in turn provides a professional service to the public. To help achieve this the organisation has implemented and developed personnel policies and procedures to ensure fairness and consistency in its recruitment, retention and deployment of staff. Treating all staff and members of the public with dignity and respect irrespective of race, belief, gender, sexual orientation, disability and age is a key principle of this approach.

During the period of this plan Merseyside Police continues to care for its staff through the provision of health, safety and welfare services to attempt to create a healthy working environment. Enhancing training through the development of a training academy with the aim of ensuring all staff has the necessary knowledge and skills to provide a professional service is very high on the agenda for Merseyside Police.

The organisation has established a set of values to help create a climate in which leaders are encouraged to lead and maintain appropriate standards of behaviour. Equally, a constant focus on performance delivery is to be maintained.

## **8.3 Neighbourhood Policing**

The Total Policing strategy established a new approach to policing in Merseyside through neighbourhood policing. The strategic aim of the organisation was to implement four strands to neighbourhood policing, which are as follows:

- Neighbourhood policing through locally based officers;
- A focus on serious and organised crime;
- Citizen focus; and
- Support departments.

The following sections of this chapter introduce how neighbourhood policing exists within Merseyside Police.

### **8.3.1 Neighbourhood Policing Through Locally Based Officers**

Merseyside Police is broken down into six BCUs, which in turn are broken down into a series of neighbourhoods. An Inspector responsible for local performance and the staff based within that area leads each neighbourhood. There are approximately forty neighbourhoods within Merseyside Police. Staff based in the neighbourhoods can be broken down in the following ways.

Neighbourhood dedicated officers are dedicated to a neighbourhood and will not be routinely abstracted for events outside their area. Dedicated officers are given the remit of community focus and are responsible for developing relationships within their area. Local hot spots of criminal activity and antisocial behaviour are to be located by dedicated officers, with intelligence fed back to command teams.

Neighbourhood patrol officers provide 24 hour calls for service response. Together, with neighbourhood support officers, they are responsible for proactive and targeted operations against persistent offenders and dealing with road policing issues.



Neighbourhood support teams are offender focused and lead the work targeted against persistent offenders.

Police Community Support Officers (PCSOs) provide a high visibility presence in local communities and take the lead with regards to tackling antisocial behaviour and reassuring the public.

The Special Constabulary provides support for the work of regular officers and is based in neighbourhood teams tackling issues identified by the local communities.

Volunteers provide a supplementary service to the work carried out by police officers and police staff.

Volunteer cadets working under appropriate supervision provide a support service in non confrontational roles whilst operating in neighbourhoods and at local public events.

The strategic aim of employing a neighbourhood policing model is to integrate with local communities. This approach ensures that officers and staff know their roles within local communities and which areas need to be focussed on through developing an understanding with the public.

### **8.3.2 Serious and Organised Crime**

It is understood that the most serious and organised forms of crime are not geographically bound within neighbourhoods. Tackling serious and organised crime in Merseyside is carried out through two strands of the organisation.

The Matrix team targets organised criminals who commit crime themselves. Offenders are typically characterised as criminals, having access to drugs and the money that comes from such a trade, who also possess illegal firearms. Matrix disrupts factions involved in the criminal use of firearms, illegal drugs and serious

robbery and other serious crimes. Matrix contains an armed support capability for deployment where and when it is required. This allows the Matrix team to target gangs that use guns to commit crime and intimidate witnesses, whilst holding them to account for their actions.

The Force Crime Operations Unit (FCOU) complements the work carried out by Matrix. The FCOU targets the top level organised criminals who are typically characterised as those who have long histories of criminal activity, who put distance between their crimes and the profits they receive from it. The activity of such organised criminals usually covers all BCUs within Merseyside Police and sometimes also extends to different Forces or countries. The FCOU targets offenders engaged in drug activity, illegal immigration, people trafficking, money laundering, serious fraud, terrorism and domestic extremism. The FCOU contains a high tech computer crime unit concerned with targeting offenders who sexually abuse children. The FCOU works closely with the Witness Protection Unit to assist threatened intimidated and vulnerable witnesses.

### **8.3.3 Citizen Focus**

The strategic aim of Merseyside Police contains total care for victims as one of its most important functions according to the Blueprint for Service Improvement. Citizen Focus draws together several departments from across the organisation to work towards achieving total care for victims.

Citizen Focus works towards recognising and understanding the needs of victims, witnesses and law abiding citizens. In terms of neighbourhood policing this means that each policing service must be made readily available to the public.

From the initial point of contact through to bringing offenders to justice, the respective departments that oversee calls and crime recording and corporate criminal justice have responsibility to manage and administer processes that support the victim, witnesses and law abiding citizen.



This approach will include how Merseyside Police deals with the initial call for assistance, and also how it supports and keeps victims and witnesses informed of progress in dealing with the incident or ongoing investigation.

Through regular public meetings, surveys and questionnaires, communication and marketing the organisation will undertake a process of monitoring the satisfaction and confidence levels of victims and local communities in the service we provide.

The information obtained helps to define policing priorities and identify those areas where improvement is required.

#### **8.3.4 Support Departments**

Support departments exist within Merseyside Police to assist the organisation fulfil its responsibilities. Through implementing support departments officers are freed of some of their responsibilities and can spend more time performing front line policing duties.

The information systems department assumes responsibility for purchasing, developing and utilising of technology and information systems to ensure operational officers have access to the necessary tools to help them. This currently includes:

- Automatic number plate recognition;
- Automatic vehicle location;
- Mobile fingerprint recognition;
- Mobile data access; and
- Recording systems.

In addition, Merseyside Police looks to exploit opportunities presented by the Internet both to engage with the public through online forums and emails, and to seek public assistance in the identification of offenders in published CCTV recovered images.

Through personnel support departments and the personnel roles undertaken in BCUs it is the organisation's ambition to have the right people with the right skills in the right places, at the right time. This is an ambitious target in any organisation but Merseyside Police has set itself such a target. Through recruitment, people management, health, welfare and development processes, it is anticipated that Merseyside Police will be resourced appropriately, embracing the benefits of diversity, equality of opportunity and fairness.

Operational support departments will continue to work alongside and support the delivery of our neighbourhood policing and tackling serious and organised crime.

Teams of specialist officers provide expertise in relation to:

- Protective services;
- Forensics;
- Air support;
- Mounted section;
- Dog handling section;
- Traffic;
- Intelligence gathering and dissemination; and
- Planned policing operations.

In addition, there is skilled staff that conducts research and analysis of the performance, undertaking the management of risks and administrative functions.

## **8.4 Continuous Improvement**

Whilst Merseyside Police wants to improve all aspects of the service provided some prioritisation of issues is necessary. This enables the organisation to improve the service offered to local communities and will help achieve the delivery of Chief Constable Bernard Hogan-Howe's vision of being the best police service in the UK.



To help achieve this consultation has been conducted with local communities who have provided information on what issues are of concern to them. In addition, the Force, and the Police Authority, to ensure that it achieves the performance targets set at the beginning of each financial year monitors performance across a range of statutory performance indicators. Where the Force has failed to meet targets, measures will be introduced to ensure performance improves to the extent that when compared to most similar Forces performance in each statutory performance indicator Merseyside Police will be ranked first.

The Police Authority has summarised the key issues contained within a 'Local Policing in Merseyside' plan [183]. This was distributed to every household on Merseyside and gave recipients the opportunity to indicate if they agreed with proposed policing priorities and to suggest alternative priorities for consideration in the future.

Whilst the majority of respondents agreed with the local policing priorities that had been set, there were other suggestions, with the most common being:

- Increase police visibility;
- Improve call handling and response;
- Tackling organised crime and drugs;
- Improve roads policing; and
- Educate youths who cause antisocial behaviour.

In addition to the findings of the many surveys, questionnaires and meetings that both the Police Authority and Force arranged the feedback received from this process has helped to identify the priority areas where improvements are required.

## **8.5 Developing an Understanding of the Problem Situation**

Whilst the Force has shown ambition through developing its Total Policing approach it must be appreciated that this is a long term aim, with several difficulties to be overcome before this is achieved.

The National Intelligence Model (NIM) produced in 2005 [184] provides details of how Forces in England and Wales should be structured to ensure that consistent policing is offered to the public. However, the structures implemented by NIM lead to a situation arising whereby management level officers are unable to conduct any front line duties, and potentially more importantly, have little or no interaction with officers who actually work on the streets. Because of this there is a significant gap between front line officers and command teams across the Force. This inevitably has an impact of the level of service offered to the public.

The Blueprint for Service Improvement aims to be implemented before the end of 2008 but before this can happen some difficulties, such as the gap between front line officers and command teams must be overcome.

There appears to be a significant problem of information starvation within Merseyside Police. This presents itself through officers attempting to conduct duties associated with their role without being able to access the information they require. This can lead to difficulties with decision making and is potentially dangerous when considering the type of duties performed by front line officers. Currently senior managers are also unable to gather the information required to brief staff at the beginning of shifts, which only serves to provoke the situation.

Such a problem has led to officers collecting some information themselves, which is either irrelevant or inappropriate, and keeping it to themselves. Other officers prefer to contact civilian support staff and request that they provide information, which is not necessarily their role and leads to friction between the two. The problem of information starvation, which itself must be overcome, has led to a cultural problem within the organisation too.

The two problems outlined above create an environment in which it is difficult for the organisation to move forward and achieve its ambition of fulfilling the Blueprint for Service Improvement and becoming the best Force in the UK. In recent years IT solutions have been implemented to resolve the situation, however, these have not proved to be successful as staff felt their views had not been taken into consideration and that it was yet another system they would have to learn to use.



Staff became disillusioned with having to develop skills for so many IT systems where they would record information but not be able to retrieve it when it was required.

In addition to this problem Government led initiatives and law changes are frequently imposed on Merseyside Police with little or no advanced notification. Given the organisational difficulties being experienced it is almost impossible to keep abreast of such developments. Changes are required to ensure that future changes can be accommodated for. To achieve this Merseyside Police must have implemented a resolution to its internal problems first.

Despite the difficulties that have been discussed above it is possible for Merseyside Police to develop itself and implement suitable resolutions in order to achieve the ambitious targets identified in both the Total Policing strategy and the Blueprint for Service Improvement.

## **8.6 Summary of Introduction to Case Study Organisation**

An information starvation problem exists whereby staff cannot collect the information they require to perform their duties properly. This presents itself in many different ways such as inefficient practices leading to longer investigative periods, misinformed staff, and a level of service offered to the public that requires improvement.

A cultural problem exists where there is tension between officers and civilian support staff. Whilst it can be understood how the tensions came into existence it does not help the organisation take the necessary steps towards improvement and therefore must be overcome. IT solutions that have been implemented in recent years have been deemed as failures due to the cultural problems that exist and the introduction of NIM has provoked the situation further.

Despite the problems being experienced in Merseyside Police the organisation, and its staff, is keen to implement suitable resolutions and continue its drive towards

achieving the aims of the Total Policing strategy and the Blueprint for Service Improvement. Fundamental to the organisation is the principle of continuous improvement, which is something that will become possible should a suitable resolution be implemented

## **8.7 Application of Generic Framework to Case Study Organisation**

The generic framework will be applied to the problem situation, as described below, in an effort to resolve problems that the organisation has been experiencing. This study will follow the application of each of the seven Steps of the generic framework.

### **8.7.1 Generic Framework Step 1**

Merseyside Police is the fourth largest police Force in the UK. By 2008, when Liverpool will be the Capital of Culture, Merseyside Police is expected to have expanded to 9,000 staff. The current Chief Constable, Bernard Hogan-Howe has pledged to make Merseyside Police the best in the UK and has created a Blueprint for Service Improvement which outlines what he intends to achieve by 2008.

Due to the nature of the policing environment staff are scattered across the constabulary. The central command of Merseyside Police is based at the Headquarters building in Liverpool city centre. The actual 'front line' policing is carried out by staff who are based in stations throughout the constabulary. The current NIM attempts to bridge the gap between the central command and front line officers by assigning BCU Commanders and Neighbourhood Inspectors to areas across the constabulary. However, the BCU Commanders and Neighbourhood Inspectors do not have front line duties and are now effectively occupy management roles. Because of this there is still a gap between front line officers and their command teams. Front line officers are unable to gather the information they require themselves, which has an effect on the level of service offered to the public.



The Blueprint for Service Improvement is to be implemented by 2008 and in order for this to succeed changes to Merseyside Police must take place. As mentioned above, there are limitations to NIM, which must be overcome and the culture within Merseyside Police must also be addressed. At present staff are unsure who to contact when attempting to gather information and the processes in place are somewhat ad hoc. In addition to this the culture in Merseyside Police is that officers expect civilian support staff to provide them with the information they require. In the modern policing environment this is not possible and a system needs to be put in place that will see a shift in culture whereby officers can retrieve their own information. At present there are limitations as to which information can be provided to front line officers. For example, only custody officers can retrieve custody information and so on. Although there are instances where information must be kept from front line officers (such as CID drug investigations) it is essential that everyone walking the streets is fully briefed.

It is my intention to analyse the situation and implement a solution that resolves the problems and is able to adapt to changes in the modern policing environment. By implementing the solution Merseyside Police should be able to cope with its expansion over the coming months and years, a shift in the internal culture and provision of information to officers in order to facilitate the Chief Constable's Blueprint for Service Improvement to the public.

### **8.7.2 Generic Framework Step 2**

Before analysis of Merseyside Police can begin roles for the main players in the evaluation have to be defined. These are as follows:

- Client – Merseyside Police has identified the need to analyse the current situation.
- Problem Solver – The analyst working with the generic framework within Merseyside Police.
- Problem Owner – Both officers and civilians in Merseyside Police.

In order to gain a full understanding of the problem situation several months of speaking to staff from all parts of Merseyside Police has taken place. This has involved conducting interviews, sitting in meetings, analysing the information flows, becoming familiar with Force databases and the Intranet whilst patrolling with front line officers to see how they do their jobs.

It appears that the main overriding problem in Merseyside Police is communication. This is having an effect on the service offered to the public due to front line officers patrolling the streets without sufficient information. This affects the amount of time it takes to detect crime, delays in producing documentation for court appearances and even contacting victims of crime to keep them informed of case progress. The NIM was successful in that it implemented what on paper was a sound hierarchical structure, however the culture in Merseyside Police was not taken into account. This has led to the introduction of systems that record incredible amounts of information but no thought was given to how staff can retrieve it themselves. This is something that affects everyone in Merseyside Police.

By introducing a new information system that the staff feel will assist them in their jobs, so that they can get the information they need when it is required, this part of the communication problem can be removed. This is very much a primary task problem as it has an effect on the service offered to the public, which is why Merseyside Police is in existence in the first place.

It should be noted at this point that there are two cultural problems. Initially the internal cultural problems in Merseyside Police have been discussed. However, there are external agencies which can directly influence the running of Merseyside Police. The Home Office is responsible for releasing changes in the law and Merseyside Police has to comply with any constraints they impose. Secondly, the Merseyside Police Authority (MPA) provides Merseyside Police with their annual budget and dictates staffing levels. In both cases the decisions taken by public servants with no police background directly affect what happens to Merseyside Police. Although the Chief Constable is on the board of MPA he can be outvoted on policing matters.



However, it is felt that to be in a position to attempt to change the culture at the Home Office and MPA it is a prerequisite that Merseyside Police is in a position culturally to cope with the changes it is about to impose on itself before 2008.

Systems, which have either been verbally or IT operated, have been unsuccessfully introduced in recent years. The major reason for the failure of these systems is that the customers, usually the front line officers, were not asked for their requirements. The command teams above them decided what should be incorporated into these systems on their behalf.

At present there are no set procedures or standardised approaches for implementing systems aimed at assisting Merseyside Police staff. As a consequence this leads to information being gathered on an ad hoc basis. This approach relies on using contacts that staff have stumbled across whilst looking for relevant information.

Due to the nature of the modern policing environment it is accepted that whilst some investigative work is being carried out it is not possible to share certain information with other Merseyside Police staff. Once cases have been closed the information recorded should be available for retrieval, although this is not always the case for legal reasons. Research so far suggests that there is a 'secret squirrel' culture in some parts of the organisation whereby staff are unwilling to share the information they have recorded as they do not have complete confidence in the Force systems.

The issue based problem at Merseyside Police cannot be seen or felt, although the results of the problem can be noticed through lack of informed staff, and therefore the introduction of new technology cannot necessarily resolve this part of the situation.

At Merseyside Police the separate departments try to become as self sufficient as possible because they feel that they are in an isolated position. The intention of NIM, by introducing BCU Commanders and Neighbourhood Inspectors, was to try to prevent this from happening by drawing the staff closer together in an attempt to create a cooperative environment which benefits the organisation.

The gap between central command at Headquarters and front line officers still exists despite the introduction of NIM. Although each area of the constabulary has its own BCU Commander and Neighbourhood Inspector now the NIM has not addressed the cultural problems in Merseyside Police.

When new systems have been implemented, in particular IT systems, the requirements of front line officers have not been considered.

There is some tension between officers and civilians. Officers have always felt that the civilians should provide them with the information they require. However, this is not possible in the modern policing environment and officers must take responsibility in the provision of information.

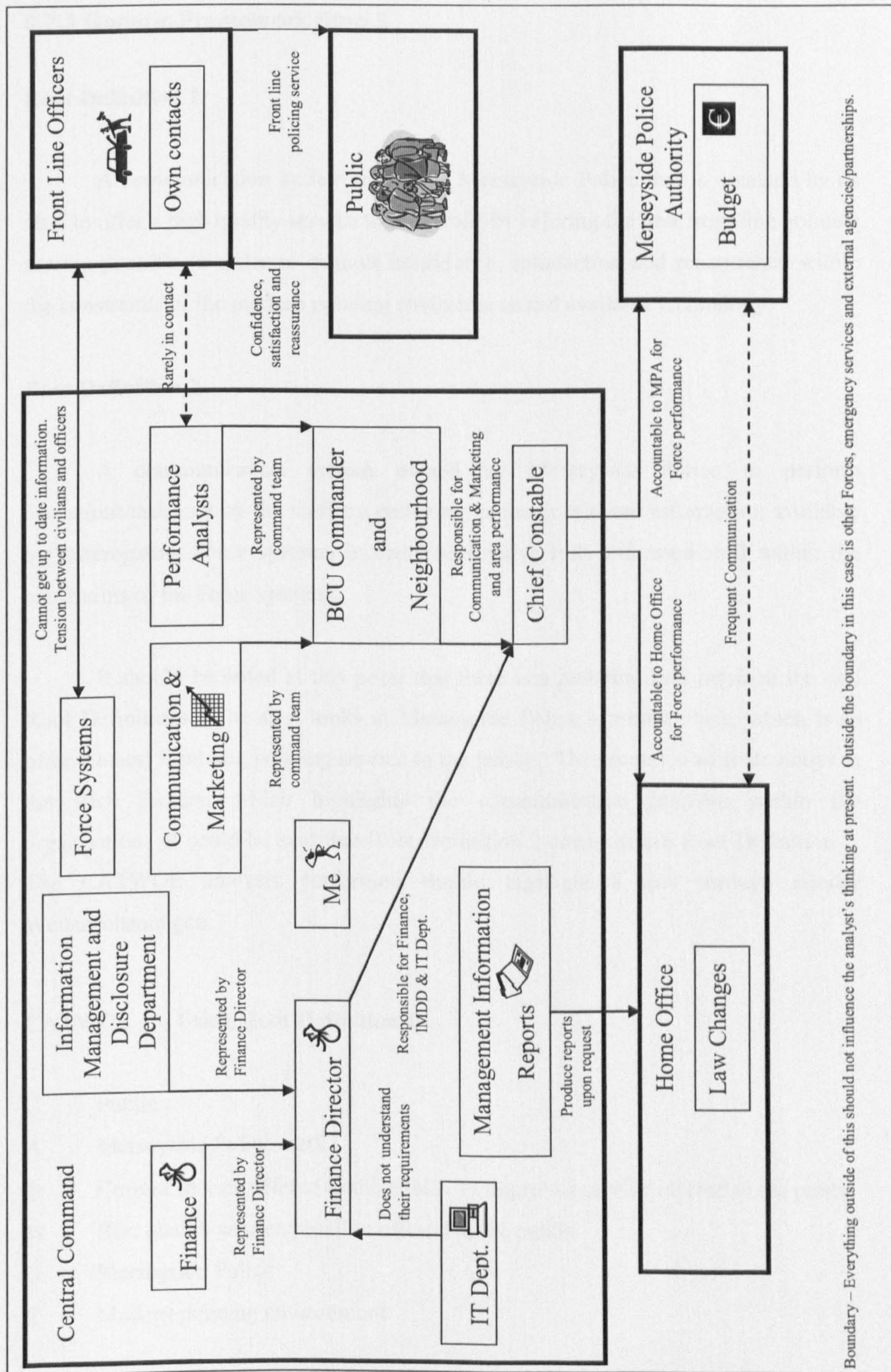
Requirements are constantly changing and due to the current culture the officers always feel that they are 'fire fighting'. The current inefficient practices mean that it takes longer to detect crimes, delays in producing documentation for court cases and contacting victims of crime to update them on case progress.

There are two main problem areas in Merseyside Police. Firstly, the lack of information provided to front line officers is hindering them whilst they try to carry out various duties. Secondly, is the culture within Merseyside Police, which has seen a gap open up between strategic commanders and front line officers despite the introduction of NIM. The two problems are certainly not mutually exclusive as communication is a major factor in both of them.

A Rich Picture of the problem situation has been developed, which is shown below.



# Rich Picture of the problem situation within Merseyside Police



Boundary – Everything outside of this should not influence the analyst's thinking at present. Outside the boundary in this case is other Forces, emergency services and external agencies/partnerships.



### **8.7.3 Generic Framework Step 3**

#### **Root Definition 1**

A communication system owned by Merseyside Police that is operated by its staff to offer a high quality service to the public by offering the best front line policing service possible in order to achieve confidence, satisfaction and reassurance within the constraints of the modern policing environment and available technology.

#### **Root Definition 2**

A communication system owned by Merseyside Police to perform operations/tasks set by the staff by gathering as much relevant information available by interrogating Force systems in order to achieve fully informed staff within the constraints of the Force systems.

It should be noted at this point that there is a potential link between the two Root Definitions. The first looks at Merseyside Police's primary task, which is to offer the best front line policing service to the public. The second is an issue raised in the Rich Picture, which highlights the communication problem within the organisation. It could be said that Root Definition 2 exists within Root Definition 1. The CATWOE analysis performed should highlight a link through similar Weltanschauungen.

#### **CATWOE 1 – Using Root Definition 1**

C	Public
A	Merseyside Police staff
T	Current service offered to the public → improved service offered to the public
W	Best quality service possible offered to the public
O	Merseyside Police
E	Modern policing environment



**CATWOE 2 – Using Root Definition 2**

- C Merseyside Police staff
- A Merseyside Police systems
- T Current information available to staff → improved information available to staff via improved information retrieval system
- W Fully informed staff that can offer the best possible service to the public
- O Merseyside Police
- E Capabilities of the staff and Force systems

**8.7.4 Generic Framework Step 4**

Whilst working through the seven Steps of the generic framework in order to establish a structured understanding of the problem situation and implement an appropriate resolution it can be seen that Merseyside Police is capable of offering the public a better service than it currently is. The first step towards ensuring that this situation is improved upon begins with embedding a suitable Conceptual Model within Step 4. The figures below provide the high level functionality of the Conceptual Model and then a demonstration of the one provided in the Merseyside Police case study.

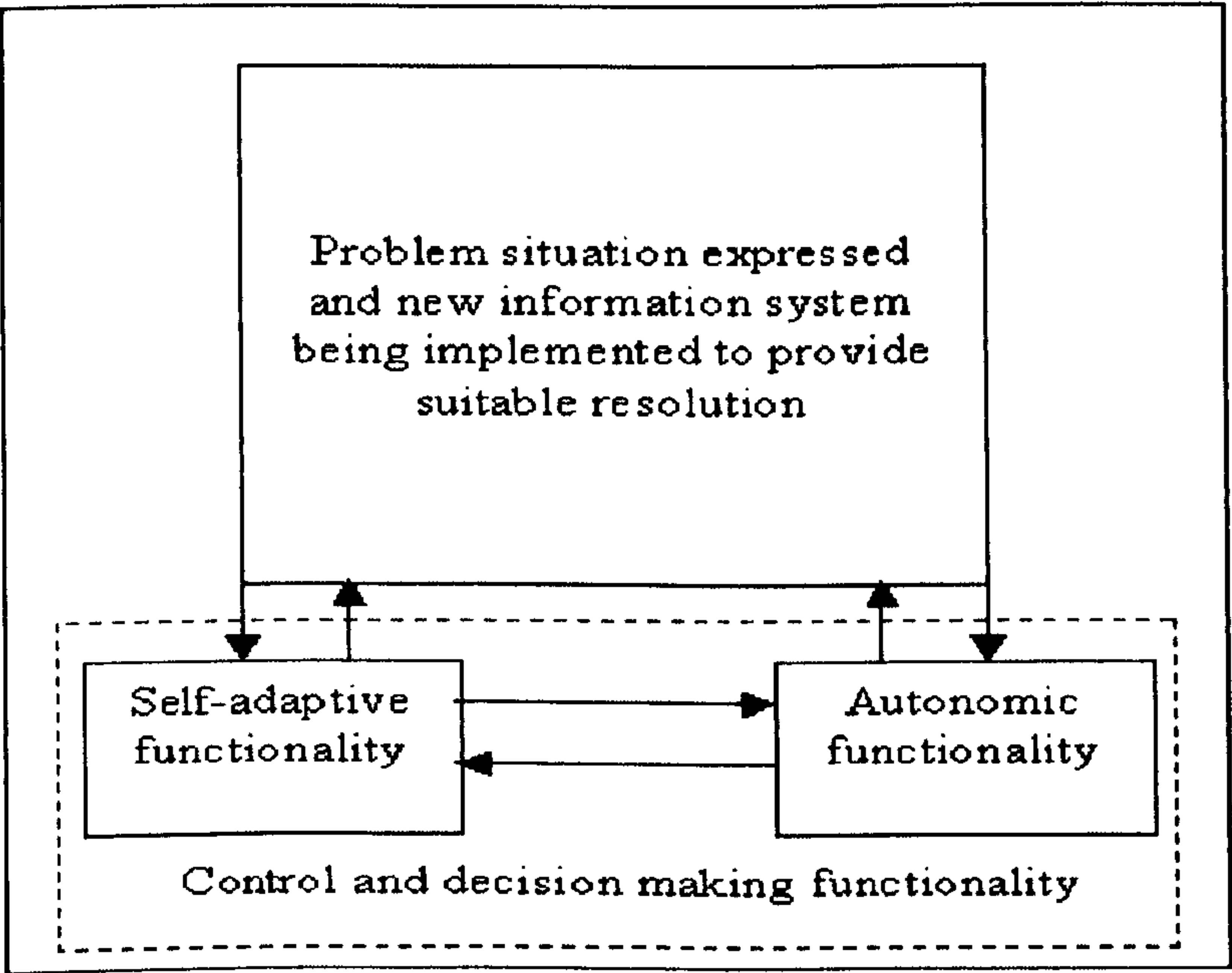


Figure 25: High level view of the conceptual model with self-adaptive and autonomic functionality.

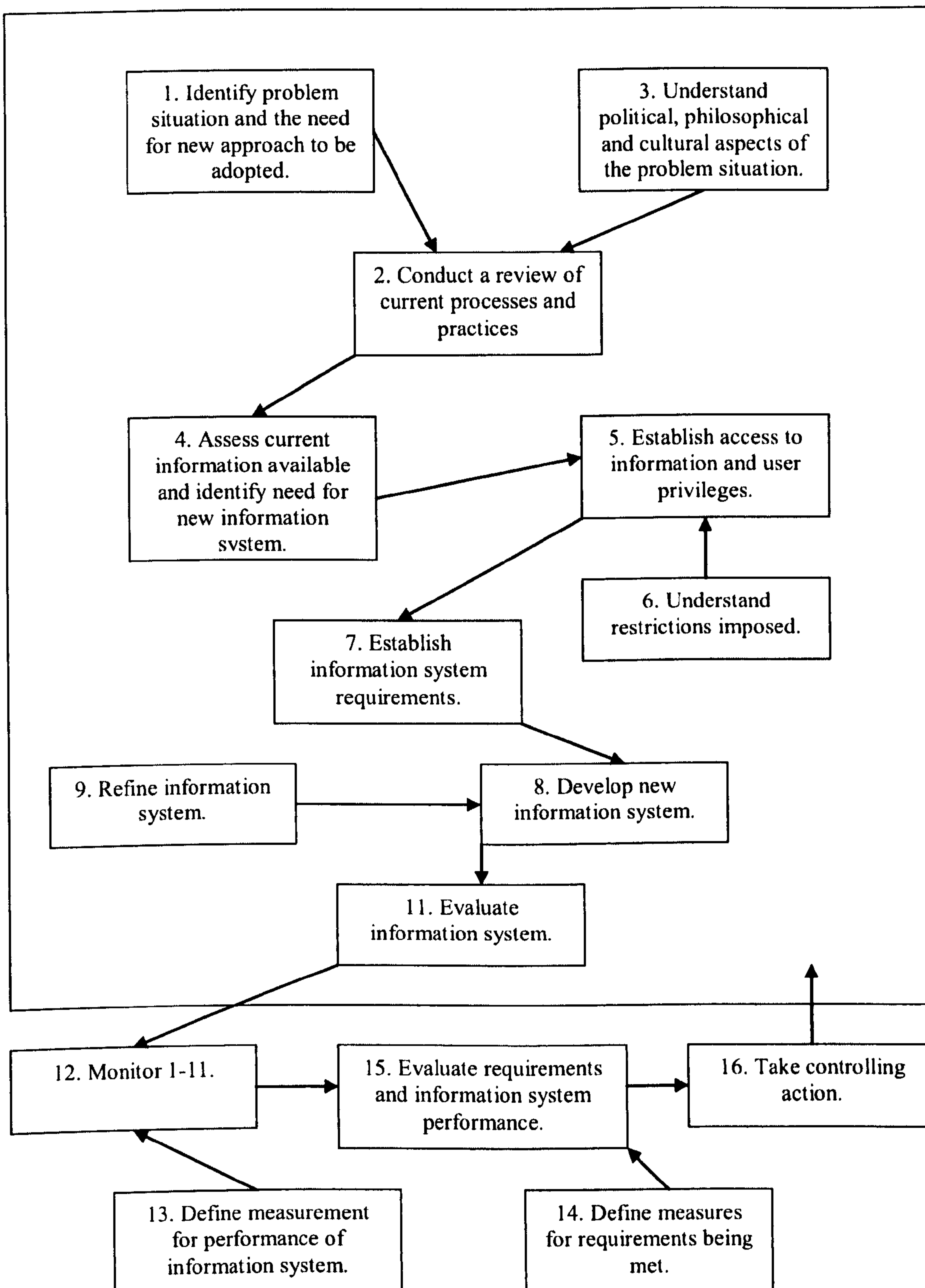


Figure 26: Conceptual model of Merseyside Police case study.

The Conceptual Model works at quite a high level in this instance as it covers the whole ‘ideal scenario’ to improving the problem situation within Merseyside Police. There are several levels of recursion below this Conceptual Model, which work in much the same way as System 1s from VSM. Activities 12 to 16, as shown



above, form the basis of the monitoring functionality that must exist within the information system to ensure that it is capable of adapting to change.

Step 5 of the generic framework will determine how the management functionality (Systems 3, 4 and 5) will operate.

### **8.7.5 Generic Framework Step 5**

The application of R<sup>5</sup> will lead to the definition of the ‘hows’ of the problem situation within Merseyside Police. In this case R<sup>5</sup> will be applied to lead towards developing suitable resolutions for the communication problem that is currently in existence.

#### **8.7.5.1 Resources**

Resourcing, as with all public sector organisations, is one of the main restrictions on any work that is carried out. Given the current level of financing the opportunity to recruit additional front line officers is not available to Merseyside Police. However, this does not mean that the service to the public cannot be improved through better communication within the organisation. Significant improvements can be made without needing to recruit additional front line officers.

For the purposes of this case study the main resource implications involve Merseyside Police staff. A major part of the communication problem is that staff is scattered across the constabulary with the Force command team based centrally. This is something that cannot be overcome due to the nature of policing. Crimes occur across the constabulary and therefore officers need to be stationed appropriately rather than based in one central location. The resolution will therefore need to take into consideration that staff will be working from different locations across the Force and that if an IT application formed part of the selected resolution that it could work within the limitations of the organisation’s technology.

### **8.7.5.2 Rules**

To ensure that the organisation complies with various strict codes and ethics there is a significant number of rules in place. These rules are not set by the Home Office or the MPA but are established by Merseyside Police in an attempt to offer a high level of service to the public. These rules include contacting victims of crime to inform them of case progression and whether they require any additional support. Each BCU within the Force applies rules in a different to ensure that the service offered is tailored to the public's local needs.

The Force also sets itself some performance indicators that are applied in the same way across the constabulary. These relate to managing crime levels, detections and dealing with antisocial behaviour. Ensuring that the service offered to the public through the implementation of a new information system is crucial.

### **8.7.5.3 Responsibilities**

The organisation has a strict hierarchical structure for both officers and civilian support staff. With the rank structure comes a responsibility depending on your position within the organisation. At each rank within the organisation there is also a requirement for staff to gain access to different information too. Any information system that is developed must take into consideration the cultural aspects of the rank structure and who takes responsibility for different parts of Force performance. The information system will also need to tackle the cultural tensions between officers and civilian support staff.

The responsibilities for different parts of Force performance will also lead to a requirement to restrict access to certain information. It is imperative to the organisation that the intended improved communication within the Force does not see a shift from staff struggling to gather relevant information to having far too much irrelevant information. Any new information system that is created through the generic framework must adopt responsibility for the provision of information. Misinforming staff is a far more serious problem than not informing staff as the



service offered to the public would be severely compromised. As a consequence, the staff carries responsibility, depending on their rank in the hierarchical organisation, and the new information system itself has responsibility for assisting staff achieve the objectives that they are responsible for.

#### **8.7.5.4 Regulations**

The Home Office and the MPA strictly govern the activities and functions carried out by Merseyside Police. There are regulations imposed upon the Force which it has no control over and can only adjust to any changes that occur. This usually takes the form of changing Home Office requirements for crime counting rules or processes that take place in the criminal justice system.

Whatever changes are experienced it is imperative that a new information system must respond to these changes within given timescales and boundaries. Communication with both internal and external environments at this stage is absolutely crucial to the information system's success. If a failure or a conflict is collected here it could lead to serious damage with regards to the service offered to the public and communication within the Force.

#### **8.7.5.5 Recommendations**

Recommendations for the new information system must take cognisance of the restrictions imposed on it, along with a full appreciation of the aspects of the Force that it will be attempting to implement a resolution for. To this end, an information system must take into account the service offered to the public and the communication problem that exists within Merseyside Police at present.

There is no scope for additional staff, and no financial backing to purchase new technology. As a consequence, the new information system must not lead to an increase in the workload of staff and, if an IT solution was deemed the most appropriate, must not require any technology that was beyond the organisation's current capability.

The hierarchical structure within Merseyside Police must be understood and appreciated, although it seems clear that changes must take place to improve the current communication and cultural problems that exist. Staff must see the potential benefits of the new information system, if the culture of being provided with information is to be changed to staff gathering it themselves, and the responsibilities of ranks and various access levels must be fully incorporated. This approach will also ensure full compliance with the Data Protection Act and adherence to the Management of Police Information Act.

#### **8.7.6 Generic Framework Step 6**

The R<sup>5</sup> analysis from the previous Step shows how recommendations for potential resolutions can be derived. The analysis also conducts a comparison between Step 4 and the real world problem situation that Merseyside Police is experiencing.

It can be seen that there are some rules and regulations in place that cannot be altered. However, there are parts of the organisational culture and communication problems that can be improved. It is understood that staff must work in different locations across the constabulary to offer an equal service to the public. A resolution to this would be introducing a new information system which ensures that the staff across the Force can use one 'repository' for collecting the information they require. This may take the form of a new system or IT application.

If one single 'repository' was used this would be the first step towards combating the communication problem within the organisation. The culture of keeping contacts and adopting an ad hoc approach to gathering information will be superseded by this new information system.

Merseyside Police has always struggled to encourage officers to use new systems, whether they are verbally operated, paper based or IT based as they see them as being "yet another thing to do" and something to be avoided at all costs. This approach has a negative on the service offered to the public. By incorporating officers



views it is expected that this cultural problem will be overcome. Historically it has been senior staff deciding on everybody's behalf what requirements are to be taken into consideration. For this reason staff from all ranks will be consulted to ensure that a full appreciation of the information system's requirements are gathered.

By implementing an information system, in which staff have been consulted during the development stage, the tension between officers and civilian support staff should be eased. The information system implemented should be developed in such a way so as to allow officers to brief themselves by providing them with relevant information. Not only would officers briefing themselves see a change in police culture from the current practice it also will lead to an improved service offered to the public through better informed front line officers.

This approach will also ease the pressure on civilian support staff meaning that they can change from working in the current 'fire fighting' way to being able to plan towards the future. In turn, the frequent changes imposed on all UK Forces by the Home Office can be dealt with considerably more efficiently by civilian support staff as their time will not be filled up anymore by providing officers with information.

#### **8.7.7 Generic Framework Step 7**

Step 6 discusses which changes are desirable and feasible in the context of a new information system. It has also been discussed that there are some aspects which cannot be changed as they are beyond the control of Merseyside Police, or any external organisation that they can influence. However, this does not mean that Merseyside Police cannot reconfigure itself to accommodate these restrictions to operate in a more effective way and therefore provide a better service to the public through improved communication.

As discussed in Step 6 Merseyside Police staff cannot be based in one central location as it is required to maintain a presence across the constabulary. Therefore the most appropriate resolution would be to ensure that the information officers require can be located at one central 'repository'. This immediately resolve the problem

whereby officers do not know who to contact regarding the collection of information, and also keeping contacts hidden from colleagues.

By taking officer's requirements into account the information system is much more likely to provide them with the information that is relevant to them, which will help them carry out their day to day duties. Officers gathering their own information represents a huge cultural change within Merseyside Police, however, the potential benefits to the public through better informed officers is significant. This will also ensure that the organisation develops a resilience to change and can accommodate changes that are imposed upon by the external environment.

The development of an information system requires communicating with staff from all parts of the Force to establish their requirements. Questionnaires and meetings were arranged to collect the views of staff and the concerns that they may have in relation to a new information system.

Having communicated with the staff the next stage is to determine what is the best way of implementing an information system that conforms to the strict rules and regulations imposed upon it. Analysis of the problem situation determined that an IT solution would be the most appropriate in this context as this could provide a suitable basis for the central 'repository'. However, staff requirements need to be taken into consideration to ensure that the IT solution would be easy to use and on such a platform so as to be used by officers when they are performing their duties.

A level of self-adaptation and autonomy is to be built into the information system to ensure that should changes occur that it is able to adapt to these changes within becoming a drain on Merseyside Police's already stretched resources.

## **8.8 The New Information System**

Having made the first run through the generic framework it was time to implement the new information system. As discussed previously in this chapter the concept of a central 'repository' of information would be the first step towards



introducing the new information system. This resolution would mean the development of an IT solution that could be used by officers whilst they carry out their day to day duties.

Taking into consideration the findings drawn from running through the generic framework the requirements for the information system, and what information to include within it, were compiled. Because the introduction of the information system would see a huge change in the Force culture it was determined that the IT solution should start off by providing limited amounts of information and then slowly develop as officers became more confident. Providing too much information, even if it was all relevant, would only succeed in deterring officers from using the it. As a consequence, it was determined that daily crime and detection reports, which were manually disseminated each morning by civilian support staff, would be the core of the information used in the new information system. This would see officers use information that they are fairly familiar and comfortable with but gather it themselves. To accommodate this a process of automating the daily crime and detection reports was started. Taking into account staff shift patterns it was decided that the automated reports should be run and published at 07:00 each day.

Whilst the decision to introduce the information system, through an IT solution, was limiting the amount of information available initially it was determined that full access privileges and restrictions needed to be imposed from the beginning. This decision was taken for the following two reasons; Firstly, officers would be overwhelmed if the information provided covered the entire Force when they only required information for their particular part of the organisation. Secondly, the Data Protection Act and the Management of Police Information Act had to be enforced.

The full self-adaptive and autonomic concepts within the generic framework were to play a significant role in organising the provision of information and restricting access where necessary. The IT solution would be developed in such a way that when officers logged in their credentials would be checked against Force systems to determine their access privileges. For example, Neighbourhood Inspectors would require access to different information than their front line officers and so on. Whenever officers were posted to different parts of the Force, or were promoted, the

system would need to recognise this and take the necessary actions to ensure that the officer was provided with relevant information in their new post. This process would be carried out by the IT solution and not require constant administration by civilian support staff.

### **8.8.1 Implementing the New Information System**

At the time the case study was instigated the organisation was looking at the possibility of replacing its current crime recording system with a new one. The Chief Officers were aware that the organisation was being involved in a case study and requested that the potential for a new crime recording system be included as part of the scope of the research. Following the analysis discussed in the previous sections of this chapter the recommendation was made that rather than procuring another crime recording system the organisation should adopt a more holistic record management system. For the staff in Merseyside Police this represented a large shift in culture, however, the potential benefits of recording more than one area of business practice on one system could be understood. The current custody and case file preparation system, called Niche RMS, was selected as the most appropriate system for recording crimes on. This would therefore reduce double keying and allow officers to track a record through from the point of arrest all the way to a court result. This was met with approval by the organisation and a process of implementation was instigated. The findings of this implementation are discussed in the following chapters.

The implementation of the new information system within Merseyside Police saw three distinct levels of access introduced to the organisation. The access privileges were linked to the relevant member of staffs' role in the organisation, which would be updated without requiring manual administration. The new information system contains the capacity to update the access privileges based on information gathered from several different sources dynamically. Three levels of access have been defined to meet the needs of the organisation.

Management information access allows users to view performance related information but not transactional level record details.



Report access allows users to views restricted amounts of transactional level record details and perform some searches on this information.

Full record access allows users to access full record details and, during a later iteration of the generic framework, allows staff to view information through GPRS enabled blackberries.

### **8.8.2 Management Information Access**

Users with the management information access privilege were able to collect a significantly improved amount of performance related figures. With the majority of data being geocoded through the new crime recording system it proved to be relatively simple to ensure that the information provided was pertinent to the locality of the member of staff looking for the performance related figures.

Prior to the implementation of the new information system limited amounts of management information could be supplied by civilian support staff in relation to crime, incident and public confidence and satisfaction. The new information system ensured that the following could be provided:

- Crime and detections;
- Incidents;
- Response;
- Road traffic collisions;
- Stops searches;
- Warrants;
- Arrests;
- Activity analysis;
- Criminal justice; and
- Citizen focus, including public confidence and satisfaction.



The information provided was updated automatically at varying intervals dependent upon the relevant interfaces that exist between the new information system and the source system.

A web based portal was established as through staff consultation it was deemed to be the most user friendly approach to the provision of information. In fact, this approach was deemed to be the most appropriate for all three access privileges. Through substantial user consultation the new information system was developed with a 'look and feel' that proved to be popular amongst staff. Users with the management information access level were greeted with a front screen as shown below:

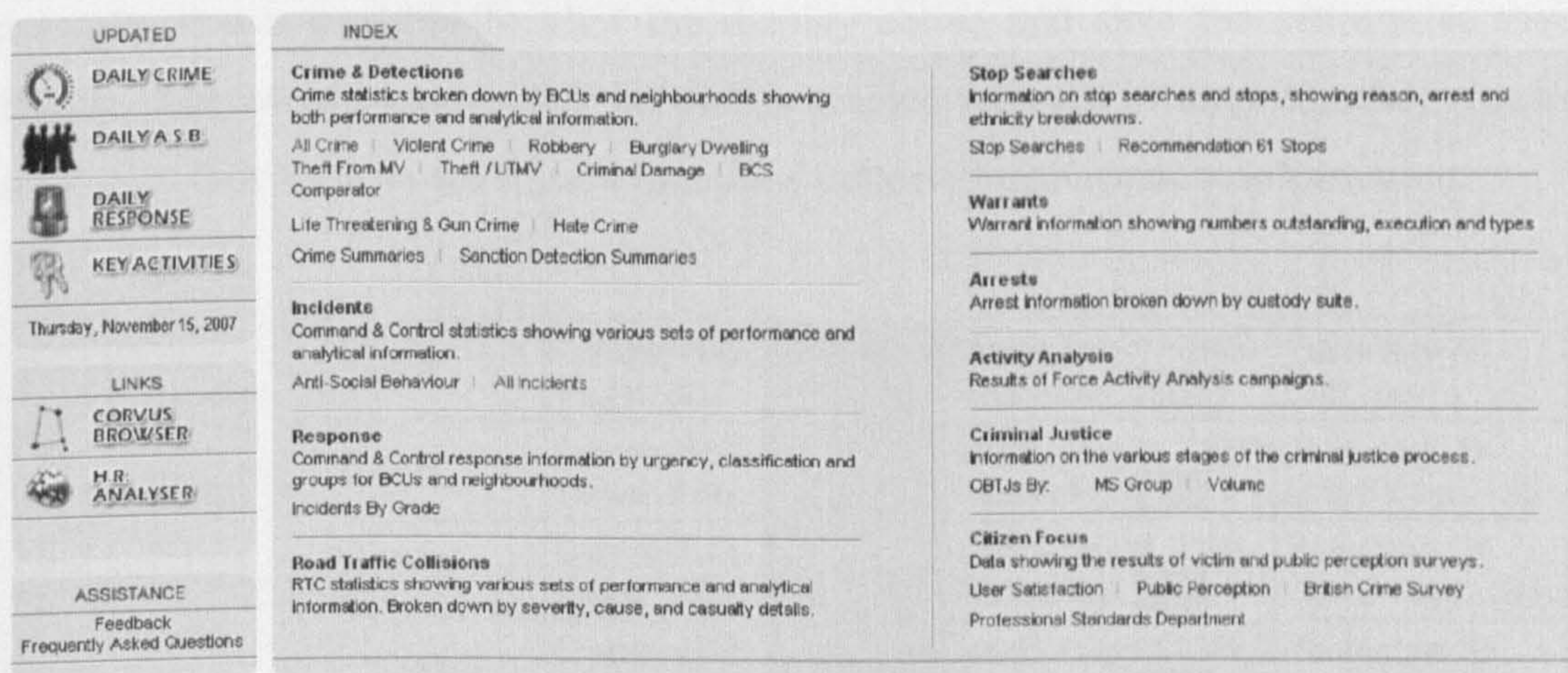


Figure 27: Screenshot of management information access.

The IT skills of staff needed to be taken into consideration when developing the new information system. As a result the new information system was developed in such a way that users with the management information access privilege only needed to click a mouse to navigate their way to the performance figures they required. This reduced the risk of collecting incorrect or irrelevant information and promoted confidence amongst users.


Users are presented with the option of displaying graphics, or deactivating them if they wish. Within certain constraints the users can tailor the way they would like the information to be displayed. On other occasions the generic framework's autonomic and self-adaptive capabilities will ensure that the new information system makes various decisions on the user's behalf with regards to information presentation.




<div><div>Home</div><div>Help</div></div>						
Crime						
SINGLE CATEGORY ▶						
SINGLE AREA ▶						
Disposal Breakdown						
SINGLE CATEGORY ▶						
SINGLE AREA ▶						
AREA & CATEGORY ▶						
Sanction Detection						
SINGLE CATEGORY ▶						
SINGLE AREA ▶						
Overviews						
NUMERIC OVERVIEW ▶						
✓ x OVERVIEW ▶						
<div><div>Set As Default</div><div>Export To Excel</div></div>						
A	Crimes: 806 Target: 862	Difference: -56 (-7%)	Month Target: 1848 Remaining: <b>1042</b>	<div><div></div><div>Actual: 57.6/day, Target: 61.6/day</div></div>	Yesterday: 51 Last 3 days: 153	
B	Crimes: 880 Target: 820	Difference: 60 (7%)	Month Target: 1758 Remaining: <b>878</b>	<div><div></div><div>Actual: 62.9/day, Target: 59.8/day, Now required: 54.9/day</div></div>	Yesterday: 54 Last 3 days: 170	
C	Crimes: 643 Target: 559	Difference: 84 (15%)	Month Target: 1198 Remaining: <b>555</b>	<div><div></div><div>Actual: 45.9/day, Target: 39.9/day, Now required: 34.7/day</div></div>	Yesterday: 25 Last 3 days: 120	
D	Crimes: 583 Target: 628	Difference: -45 (-7%)	Month Target: 1347 Remaining: <b>764</b>	<div><div></div><div>Actual: 41.6/day, Target: 44.9/day</div></div>	Yesterday: 33 Last 3 days: 106	
E	Crimes: 1647 Target: 1607	Difference: 40 (2%)	Month Target: 3445 Remaining: <b>1798</b>	<div><div></div><div>Actual: 117.6/day, Target: 114.0/day, Now required: 112.4/day</div></div>	Yesterday: 110 Last 3 days: 339	
F	Crimes: 915 Target: 763	Difference: 152 (20%)	Month Target: 1635 Remaining: <b>720</b>	<div><div></div><div>Actual: 65.4/day, Target: 54.5/day, Now required: 45.1/day</div></div>	Yesterday: 62 Last 3 days: 168	
Force	Crimes: 5474 Target: 5242	Difference: 232 (4%)	Month Target: 11233 Remaining: <b>5759</b>	<div><div></div><div>Actual: 391.0/day, Target: 374.4/day, Now required: 360.0/day</div></div>	Yesterday: 335 Last 3 days: 1056	

Figure 28: Screenshot of management information access.

Users were given the opportunity to override decisions that had been taken by the autonomic and self-adaptive functionality of the new information system. For example, it was possible to alter the display above and save the settings as their default. This would ensure that the user is greeted with the information they require in the way they want to see it each time they collected management information.

 Home

 Help

Crime

SINGLE CATEGORY ▶

SINGLE AREA ▶

Disposal Breakdown

SINGLE CATEGORY ▶

SINGLE AREA ▶

AREA & CATEGORY ▶

Sanction Detection


SINGLE CATEGORY ▶


SINGLE AREA ▶

Overviews

NUMERIC OVERVIEW ▶

✓ \* OVERVIEW ▶

 Set As Default

 Export To Excel

A	Crimes: 806 Target: 862	Difference: -56 (-7%)	Month Target: 1848 Remaining: <b>1042</b>	Yesterday: 51 Last 3 days: 153
B	Crimes: 880 Target: 820	Difference: 60 (7%)	Month Target: 1758 Remaining: <b>878</b>	Yesterday: 54 Last 3 days: 170
C	Crimes: 643 Target: 559	Difference: 84 (15%)	Month Target: 1198 Remaining: <b>555</b>	Yesterday: 25 Last 3 days: 120
D	Crimes: 583 Target: 628	Difference: -45 (-7%)	Month Target: 1347 Remaining: <b>764</b>	Yesterday: 33 Last 3 days: 106
E	Crimes: 1647 Target: 1607	Difference: 40 (2%)	Month Target: 3445 Remaining: <b>1798</b>	Yesterday: 110 Last 3 days: 339
F	Crimes: 915 Target: 763	Difference: 152 (20%)	Month Target: 1635 Remaining: <b>720</b>	Yesterday: 62 Last 3 days: 168
Force	Crimes: 5474 Target: 5242	Difference: 232 (4%)	Month Target: 11233 Remaining: <b>5759</b>	Yesterday: 335 Last 3 days: 1056

Figure 29: Screenshot of management information access.

As can be seen from the screenshot above the user has decided to remove the status bar from the management information view. This particular user has saved their settings and is greeted with this information on a daily basis.

For senior managers within the organisation, or those who are not comfortable with the information as displayed above, a graphic overview has been devised. This



was constructed following on from feedback received at the first iteration of the generic framework.

	B1	B2	B3	B4	B5	B6	B7	B
All Crime	✓	✗	✗	✗	✓	✗	✗	✗
Violent Crime	✓	✓	✗	✓	✓	✓	✗	✓
Robbery	✓	✓	✓	✓	✓	✓	✓	✓
Burglary Dwelling	✓	✗	✓	✗	✓	✓	✗	✓
UTMV/Theft of MV	✓	✓	✓	✓	✓	✓	✗	✓
Theft From MV	✗	✗	✓	✗	✓	✓	✓	✓
Criminal Damage	✗	✗	✗	✗	✗	✗	✗	✗
BCS Comparator	✓	✗	✗	✗	✓	✗	✗	✗
Vehicle Crime	✓	✗	✓	✗	✓	✓	✓	✓
Acquisitive Crime	✓	✗	✓	✓	✓	✓	✓	✓
Wounding	✓	✓	✓	✓	✓	✓	✗	✓
Common Assault	✗	✓	✓	✓	✓	✗	✗	✓
Harassment	✓	✗	✗	✓	✓	✗	✓	✗
Sanction Detections - All Crime	✓	✓	✗	✗	✗	✓	✓	✓
Sanction Detections - Violent Crime	✗	✓	✓	✗	✓	✗	✓	✗
Sanction Detections - Robbery	✓	✓	✗	✓	✓	✓	✓	✓
Sanction Detections - Burglary Dwelling	✓	✓	✓	✗	✓	✗	✗	✓
Sanction Detections - Vehicle Crime	✗	✗	✗	✗	✗	✓	✗	✗

Figure 30: Screenshot of management information access.

Users were particularly accommodating to changes taking place within the organisation as a result of them being involved in a consultation process before each change was implemented. The screenshot above was received well across the organisation as the users were benefiting from the results of something that they had requested.

For consistency, an approach which also promotes confidence in any new implementation amongst users, as many data sets as possible were given the same 'look and feel'. As shown below management information for Anti Social Behaviour (abbreviated to ASB in Merseyside Police) looks similar in layout to that of the crime information shown in the screenshots above.



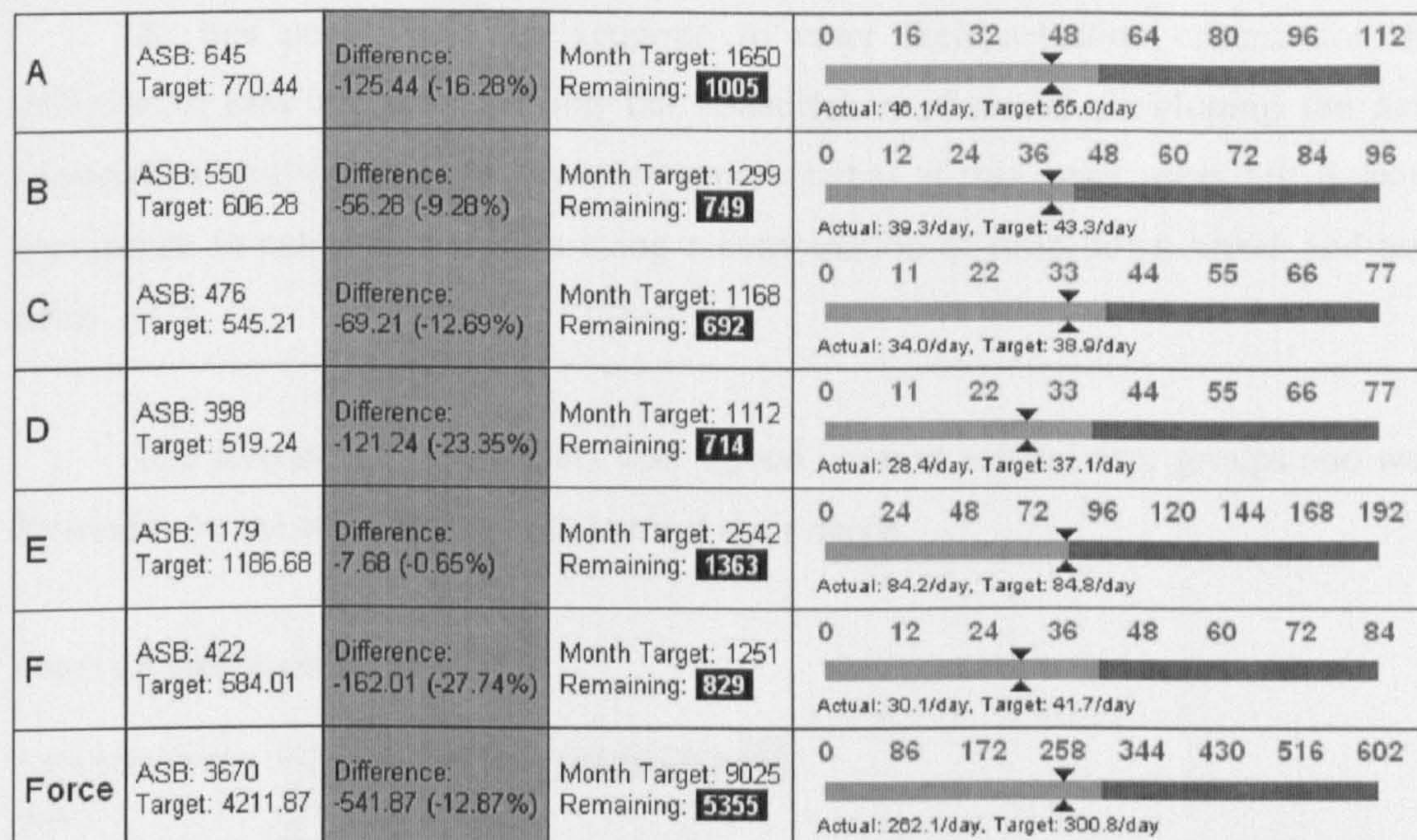


Figure 31: Screenshot of management information access.

The information provided to users allows the opportunity to drill up and down several levels of recursion to ensure that the relevant information can be collected. For all three levels of access the option of exporting the information to Microsoft Excel exists. For the management information access level users are only allowed to export aggregated data. The other two access levels allow transactional level data to be exported according to the users' privileges within certain boundaries and restrictions. For secure crimes, those which are of a particularly sensitive nature, none of the three access levels will allow users to export the record details as it could potentially compromise the investigative process along with contradicting the Data Protection Act.

### 8.8.3 Report Access


Users with report access have the same privileges as those with management information access. In addition to this, however, users are permitted to view transactional level record details with restrictions imposed on the fields of information available.





At this point users are required to enter their selection criteria for the information they require. During the consultation phase of developing the new information system it quickly became apparent that at this point users felt it more appropriate to enter their criteria using a combination of drop down boxes and text fields.


The screen that greets users was agreed upon at several user groups and was deemed to be the most appropriate to meet their needs.


**Choose Report Parameters:**

Date First Crimed: 14/11/2007 0700.15/11/2007 0659 

BCU: [ALL] 

Neighbourhood: [ALL] 

Type of crime:  

Order the results by: Earliest Date Committed 



 Run Report  Produce Printable PDF

Figure 32: Screenshot of management information access.

Users that did not wish to type in their type of crime could use a structured field through clicking on the magnifying glass icon as shown above. Similarly the user could click on the calendar control icon to select the dates and times that were relevant to them if they preferred not to type into a box. In accordance with the key requirements for the generic framework users are afforded enough flexibility to do what they want in a way that is easiest for them.

The users have been granted the ability to view the information they require through either the web based application or through an extract in Adobe PDF format.






LINKS	CRIME REF. / TYPE OF CRIME (STATUS)	MO	BEAT / CRIME ADDRESS	EARLIEST / LATEST DATE COMMITTED	INVOLVED SUSPECT(S) / OFFENDER(S)	INVOLVED VICTIM(S)
	0507091703 (NEW)	DURING PAST TWELVE YEARS OFFENDER HAS BEEN USING AN ALIAS NAME TO OBTAIN DISABILITY ALLOWANCE AND PENSION FUNDS FROM THE STATE THAT SHE IS NOT ENTITLED.	E511 / 29, PHYTHIAN STREET, LIVERPOOL, MERSEYSIDE, L61JX, , E511	01/01/1995 00:00 / 07/11/2007 08:00	- [Offender] DAVIES, THELMA 09/08/1942 (52) F (29,PHYTHIAN STREET,,LIVERPOOL,L61JX)	
	0507052520 Violence (NEW)	LINKED TO OCC-050706 6999 XXXXXXXX DOMESTIC XXXXXXXXXX OCCURED MIDWEEK/EVENING. I/P AND PARTNER ATTEND A FUNERAL. ON RETURNING TO HOME ADDRESS THEY BEGIN ARGUING. SUSPECT GRABS I/P AND THROWS HER TO THE FLOOR AND STAMPS ON HER AND PLACES HAND OVER HER FACE. SUSPECT THEN THROWS I/P AGAINST POOL TABLE CAUSING BACK INJURY. I/P SUSTAINED HAS SUSTAINED BLEEDING TO HER KIDNEY AND THERE IS A SHADOW ON HER LUNG. SHE ALSO HAS DEEP BRUISING TO BOTH ARMS AND FACE, CUT TO HER LEG AND BRUISING TO	E531 / 21, WHITBY STREET, LIVERPOOL, MERSEYSIDE, L64DH, , E531	19/07/2007 21:00 / 19/07/2007 21:30	- [Suspect] WHYTE, PAUL 02/10/1971 (35) M ( )	- [Victim] JONES, LISA CHRISTINE 27/12/1976 (30) F (21,WHITBY STREET,,LIVERPOOL,L64DH)

Figure 33: Screenshot of management information access.



The screenshot above displays some of the fields of information available to users with report access. Whilst it is important to note that the information displayed is test data, and not live data, it should be remembered that full functionality is still available.

With the information being geocoded it allows users to view linked information. The screenshot above displays information relating to crime. Links are available to the user which display suspect and offender, victim, property, modus operandi, and officer reports associated with that crime.

Offence Record:

LINKS	Offence Ref	Offence Reported	Earliest Date Committed	Latest Date Committed	Incident Location	Street Name	Postcode	Status	Notes
	0507091703	FRAUD BY FALSE REPRESENTATION (53D) OTHER FRAUD	01 Jan 1995 0000	07 Nov 2007 0800	E511	PHYTHIAN STREET	L61JX	NEW	DURING PAST TWELVE YEARS OFFENDER HAS BEEN USING AN ALIAS NAME TO OBTAIN DISABILITY ALLOWANCE AND PENSION FUNDS FROM THE STATE THAT SHE IS NOT ENTITLED.

Suspects/Offenders related to offence:

LINKS	Nominal Role	Surname	Forename	Second Name	DOB	Age at Offence	Gender	Occupation	Place of Birth	Ethnic Appearance	Street Name	Postcode	Notes	Photograph
 	Offender	DAVIES	THELMA		09 Aug 1942 0000	52	F	PENSIONER	LIVERPOOL	1. WHITE EURO	PHYTHIAN STREET		DURING PAST TWELVE YEARS OFFENDER HAS BEEN USING AN ALIAS NAME TO OBTAIN DISABILITY ALLOWANCE AND PENSION FUNDS FROM THE STATE THAT SHE IS NOT ENTITLED.	

Victims related to offence:

LINKS	Victim Ref	Surname	Forename	Second Name	Title	Gender	Ethnic	Age at	Street	Postcode	Notes
-------	------------	---------	----------	-------------	-------	--------	--------	--------	--------	----------	-------

Figure 34: Screenshot of management information access.



As can be seen in the screenshot above there is some restricted access to information associated with the crime. This is new functionality that the organisation has found increasingly popular as a tool during the crime investigative process.

The ability to export the information to Microsoft Excel exists as with all other access levels to allow users to conduct their own analysis as and when the wish. At this point users may also extract information into Adobe PDF format, which determines the presentation of the information based upon access privileges and the sensitivity of the data being displayed.

CRIME REF / TYPE OF CRIME (STATUS)	MO	BEAT / CRIME ADDRESS	EARLIEST / LATEST DATE COMMITTED	INVOLVED SUSPECT(S) / OFFENDER(S)	INVOLVED VICTIM(S)	INVOLVED PROPERTY / VEHICLE(S)	INVESTIGATING OFFICER / REMARKS
050706248 Violence (NEW)	MALE UP LEAVES THE LOCATION STATED ALONE AND IS HIT TO THE FACE BY AN UNK. OFFENDER - HE LOSES CONSCIOUSNESS - IS PICKED UP BUT REFUSES AN AMBULANCE	E711 / RAVEN WALTON VALE, L9 MERSEYSIDE, E711	26/06/2007 00:20 / 26/06/2007 00:20		[Victim] MCMURRIE, IAN 27/10/1973 (33) M (37, ROSEWORTH AVENUE, LIVERPOOL, L9 8HG)		87556 FORREST - E711 *** CRIME *** S18 WOUNDING - AUTHORIZED BY D/S HOLMES LLPS CID
050706202 Sexual (NEW)	FEMALE IN A PUBLIC HOUSE SMOKING AREA WITH A MALE WHO IS A FRIEND OF THE LICENSEE - THE FEMALE FINISHES HER CIGARETTE AND GETS UP TO LEAVE WHEN THE MALE BLOCKS HER WAY AND THEN PUTS HIS HAND DOWN THE BACK OF HER TROUSERS	A121 / SHEPHERDS REST, DARLINGTON STREET, EGRE-MONT, WIRRAL, MERSEYSIDE, CH44, A121	11/10/2007 22:00 / 11/10/2007 22:00		[Victim] BRIGGS, AMANDA 21/12/1981 (25) F (58, TRAFALGAR ROAD, WALLASEY, CH44 0EB)		81796 WALTON - *** CRIME RE-PORT *** INDECENT ASSAULT WITNESSED
050706032 Drugs (NEW)	MALE SEEN WALKING IN STREET SMOKING A REEFER TYPE CIGARETTE. UPON SIGHTING OF OFFICERS DISCARD STOP RECOVER REEFER ADMITS HIS FOR OWN PERSONNEL USE. STREET CAUTION ISSUED IN CUSTODY	B124 / WADHAM ROAD (2), WADHAM ROAD, BOOTLE, MERSEYSIDE, B124	12/10/2007 14:55 / 12/10/2007 14:55	[Offender] KINNEY, LIAM 02/04/1990 (17) M (121 BENEDICT STREET, BOOTLE, L20 2EL)		[SEIZED] [DRUGS / CANNABIS HERBAL] REEFER CIGARETTE ADULT CAUTION	88758 HOLMES - reefer type cig in the street not able to street caution b124 - poss suspect arrested dealt with by way of street caution
050706512 Criminal Damage/Arson (NEW)	CRIMINAL DAMAGE TO VEHICLE - THERE HAVE BEEN A FEW INCIDENTS, MAINLY IN OCTOBER 2007, OF DAMAGE TO A PS VEHICLE - IT HAS BEEN SCRATCHED ALL DOWN ONE SIDE, AND HAD FLOUR, AND KETCH-UP THROWN OVER IT AT OTHER TIMES	B623 / 7, BUTTS LANE, SOUTHPORT, MERSEYSIDE, PR88ER, B623	15/10/2007 12:00 / 15/10/2007 12:00		[Victim] HUDSON, NICOLA 15/01/1985 (22) F (7 BUTTS LANE, SOUTHPORT, PR88ER)	[DAMAGED] [CAR] - [DAMAGED] [VEHICLE ACCESSORY] [VEHICLE KEYS]	88553 CALVELEY - *** CRIME *** CRIMINAL DAMAGE TO MOTOR VEHICLE

Figure 35: Screenshot of management information access.

The information displayed in the PDF format enables users to see significant amounts of information through columns of concatenated data. The information provided remains within the boundaries and restrictions associated with the users' access privileges and can be retrieved quickly.

During the consultation phase of developing the new information system users requested that the Adobe PDF format be made available as the transactional level data can be extracted and saved for use in operational planning and briefing sessions. The benefit of the Adobe PDF format is that the information displayed is read only and cannot be edited. This ensures the integrity of the information being displayed and that restrictions imposed through the Data Protection and Management of Police Information Acts are complied with.



### 8.8.4 Full Record Access

Users with full record access have the same privileges as those with report access. In addition to this, however, users are permitted to view transactional level record details with only a few restrictions imposed on the fields of information available.

At this point users are required to enter their selection criteria for the information they require. During the consultation phase of developing the new information system it quickly became apparent that at this point users felt it more appropriate to enter their criteria using a combination of drop down boxes and text fields.

The screen that greets users was agreed upon at several users groups and was deemed to be the most appropriate to meet their needs.

Offence Details

Offence Reference:	<input type="text"/>	Occurrence Type:	<input type="text"/>
Date Offence Recorded:	<input type="text"/>	Date Offence Reported:	<input type="text"/>
Earliest Date Committed:	<input type="text"/>	Latest Date Committed:	<input type="text"/>
Detected Date:	<input type="text"/>	Result Date:	<input type="text"/>
Reporting Method:	<input type="text"/>	Status:	<input type="text"/>
Report Status:	<input type="text"/>	Area:	<input type="text"/>
Area Name:	<input type="text"/>	Neighbourhood:	<input type="text"/>
Offence Recorded:	<input type="text"/>	HQ Class:	<input type="text"/>
CCCJS Code:	<input type="text"/>	Disposal Code:	<input type="text"/>
Disposal Text:	<input type="text"/>	Disposal Group:	<input type="text"/>
Offence Group:	<input type="text"/>		

Figure 36: Screenshot of management information access.

Users that did not wish to type in their type of crime could use a structured field through clicking on the magnifying glass icon as shown above, in much the same way as staff with report access can. Similarly the user could click on the calendar control icon to select the dates and times that were relevant to them if they preferred not to type into a box. In accordance with the key requirements for the generic framework users are afforded enough flexibility to do what they want in a way that is easiest for them.



The users have been granted the ability to view the information they require through either the web based application or through an extract in Microsoft Excel format for further analysis.

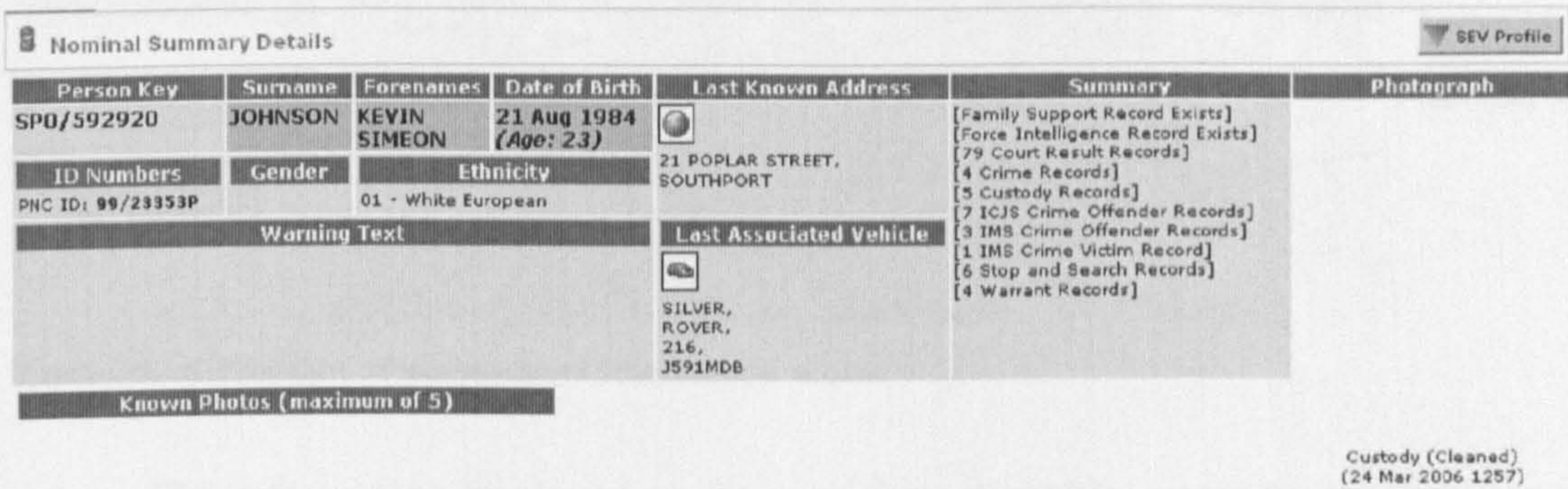


Figure 37: Screenshot of management information access.

The screenshot above displays some of the fields of information available to users with report access. Whilst it is important to note that the information displayed is test data, and not live data, it should be remembered that full functionality is still available.

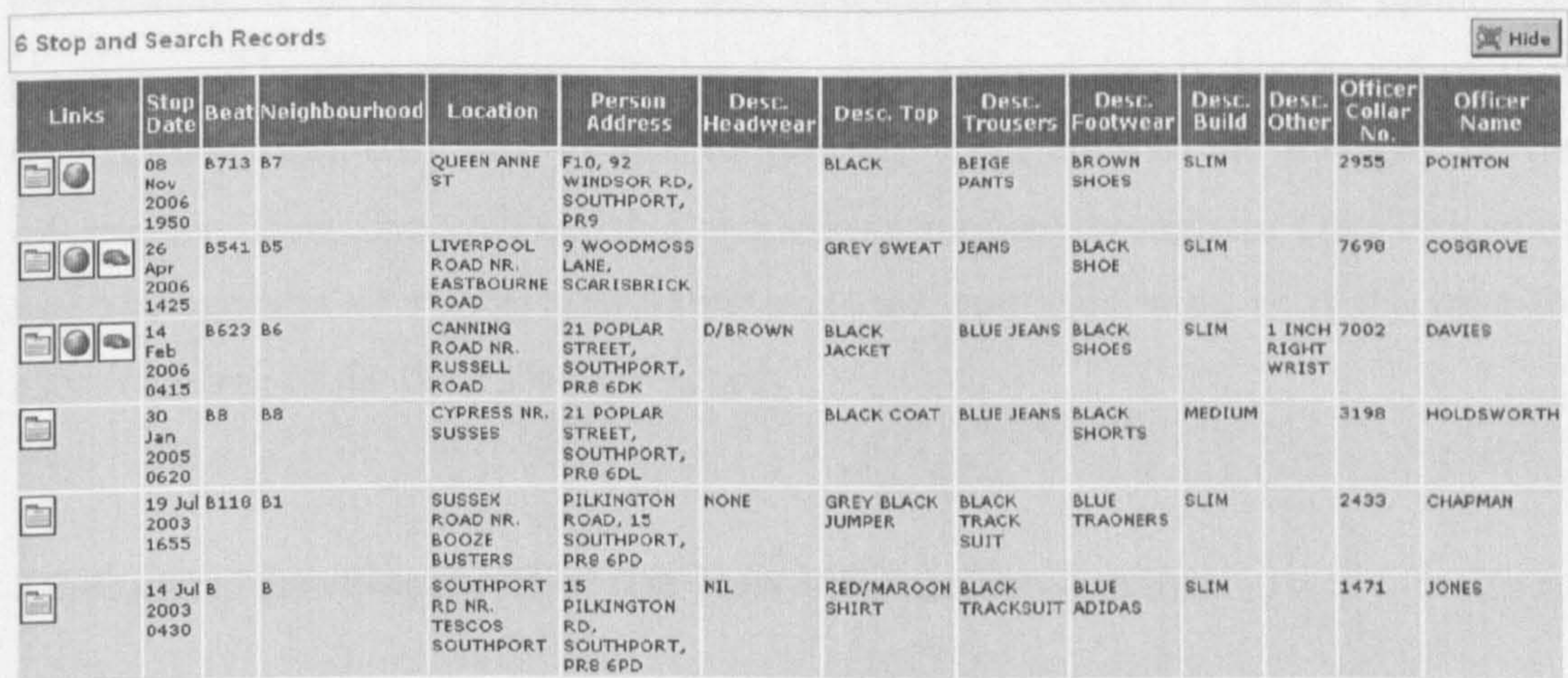


Figure 38: Screenshot of management information access.

With the information being geocoded it allows users to view linked information. The screenshot above displays information relating to a ‘single enterprise view’ of an offender and their previous record. Links are available to the user which display crime details, suspects and offenders, victims, property, modus operandi, and officer reports associated with that crime.



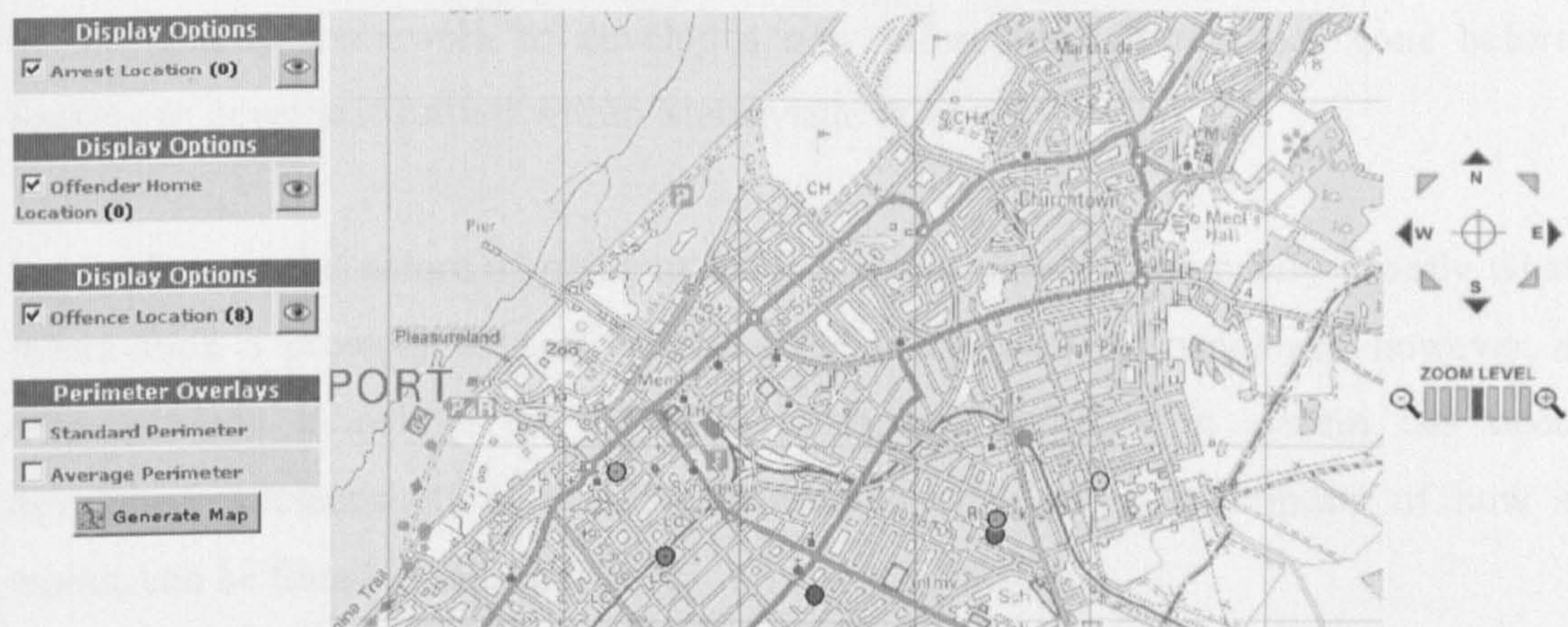


Figure 39: Screenshot of management information access.

The information displayed in the map format enables users to see significant amounts of information through interactive GIS data. The information provided remains within the boundaries and restrictions associated with the users' access privileges and can be retrieved quickly.

During the consultation phase of developing the new information system users requested that the mapping functionality be made available in addition to the transactional level data, which can be extracted and saved for use in operational planning and briefing sessions. The benefit of the mapped data is that the information displayed is read only and cannot be edited. This ensures the integrity of the information being displayed and that restrictions imposed through the Data Protection and Management of Police Information Acts are complied with, as is the case for users with any of the three levels of access.

## 8.9 Further Development of the New Information System

After the initial phase of introduction it is proposed that the generic framework reviews the successes and failures of the information system and takes the necessary corrective action. In addition to this the generic framework should look at further developing the information system in such a way that it expands its provision of information, whilst maintaining an understanding of the various cultural and communication issues that exist within Merseyside Police. However, it is deemed appropriate that a period of six months should be used to monitor how the application



of the generic framework to develop a new information system has gone before looking to develop it further within Merseyside Police.

Due to the nature of policing work it is not possible to specify exactly what information is provided to staff through their differing access privileges, however, a demonstration model of the IT solution for the information system has been developed. Screenshots of the demonstration model, and a description of how it works, can be found in the Appendix.

An evaluation of the full generic framework's implementation is in Chapter 10 of this thesis.

## **8.10 Chapter Summary**

This chapter has discussed the first implementation of the generic framework. The information system that was introduced, by means of a case study, demonstrates the application of the generic framework within Merseyside Police. It should be noted that several iterations of the generic framework will be utilised to ensure the further development of the information system to keep developing the service offered to the public and improving upon the communication and cultural aspects present within the organisation.

This chapter has showed how the generic framework can lead to a successful transition from the 'whats' to the 'hows' of problem situations and the steps that are taken to achieve the transition. The information system that was introduced in Merseyside Police will keep evolving and developing as it exists within an environment that is ever changing. The ability to respond to change through accommodating a control and management capacity and the concepts of self-adaptation and autonomy are crucial to the generic framework's success.

The following chapter establishes a platform by which the successes and failures of the seven Steps of the generic framework and the information system that was developed can be assessed.



## **9. Testing the Generic Framework**

### **9.1 Introduction to Testing of the Generic Framework**

This chapter aims at establishing a mechanism that is appropriate to ensure relevant testing is conducted. This testing has been constructed in such a way so as to facilitate a suitable evaluation of the generic framework.

An assessment of the generic framework and its adherence to its initial principles will be conducted. This assessment will highlight the successes and potential failures of the generic framework based on information gathered during the implementation phase.

An assessment of the new information system within Merseyside Police will be carried out. This will be performed to monitor whether the new information system implemented is consistent with the aims of the generic framework and whether it has improved the problem situation that existed within Merseyside Police.

Due to the vast size of Merseyside Police it was deemed appropriate that a period of at least six months was established for monitoring the effects of the application of the generic framework. With the cultural beliefs of staff within the organisation being altering through the implementation it would not be appropriate to carry out a testing phase in a shorter time period.

This chapter provides details of the testing that has been conducted along with establishing a benchmark position for the following chapter, which performs an evaluation of the generic framework and the new information system.

### **9.2 Establishing a Testing Platform**

A common practice of testing, according to EtestingHub [185] is that it is performed by an independent group of users after the functionality has been

developed but before it is released to all end users. This practice often results in the testing phase being used at the same time a project starts and is continued until that project is completed. This approach can be problematic as it can cause significant delays if a fault is highlighted part way through a project.

There are added risks in that new defects may be introduced as part of the corrections, and the original requirement can also change part way through, in which instance previous successful tests may no longer meet the requirements and will need to be respecified and redone, which is called regression testing.

To ensure that risk is minimised during the implementation and development of the generic framework and new information system within Merseyside Police a six point approach was adopted.

Requirements analysis testing should begin during the requirements phase. During the development phase users work with developers to determine which aspects of the implementation are testable and what tolerances are acceptable.

Test planning ensured that a plan was agreed upon to determine processes for testing.

Test development created an environment of example scenarios for testing are discussed to ensure full system testing is conducted, thus reducing the risk of implementing the generic framework when it has not been fully tested.

Test execution allowed the users to test the generic framework against the agreed principles and processes. The following sections of this chapter discuss how this was introduced in three phases within Merseyside Police.

Test reporting ensured an appropriate mechanism is in place to ensure that feedback is received. Based on this information the generic framework and new information system can be deemed to be successful or not with areas for potential improvement identified.



Retesting defects was incorporated into the test plan so that having identified areas for improvement an iteration of the generic framework was to be conducted with the testing process to be carried out again.

By establishing the approach detailed above it became possible to put in place a robust structure within which an accurate assessment of the generic framework, and the new information system, could be carried out.

It should be noted that not all failures must be fixed by altering the generic framework or new information system. Some failures may have been caused by errors in creating the testing platform. Some failures can be overcome through making appropriate alterations to the testing environment to bring it back in line with the initial objectives of the testing phase.

### **9.3 Establishing a Benchmark**

To ensure that an appropriate testing phase is conducted the current position of Merseyside Police must be recorded for purposes of benchmarking the effect the implementation of the generic framework, and new information system, has had on the organisation.

An analysis of current Force performance monitoring successes and failures of the implementation based on information must be gathered prior to the testing phase commencing. This has been done to ensure that no bias can affect the assessment of the implementation. This will also facilitate the development of an established benchmarked position that will remove the possibility of any preconceived ideas due to objective information having been gathered before the implementation phase began.

For the evaluation to take place the development of suitable measures must be conducted prior to the implementation of the generic framework and new information system.

### **9.3.1 Testing from a Quantitative Perspective**

Merseyside Police has an obligation to produce daily management information to ensure consistent monitoring of organisational performance. The Home Office requires monthly returns, based on the daily management information, to be provided for their assessment.

Given that performance related management information is provided to both the organisation and the Home Office it seems suitable to utilise this information for the purposes of assessing whether the implementation of the generic framework has had a positive or negative effect in this matter. The information sent to the Home Office is consistent across all Forces in England and Wales and therefore will reduce the possibility of producing quantitative information for evaluation that has any biased associated with it.

The management information provided is as follows:

- Crime levels;
- Detections;
- Antisocial behaviour levels; and
- Custody activity.

It should be worth noting that the information provided is at BCU level. This is particularly beneficial for reporting on the effect the generic framework and new information system has had as it can assess the performance of BCUs that have adopted the new approach and those that, at the point of assessment, have not during the phased implementation (which is discussed later in this chapter).

### **9.3.2 Testing from a Qualitative Perspective**

Due to the generic framework containing the capacity to cope with the softer elements of problem situations it is also appropriate to collect information from a



qualitative perspective. In essence, this is information relating to people's feelings on the successes or otherwise of the implementation. Prior to the implementation a forum of representatives for each BCU was established to ensure that feedback was collected throughout the whole process. The forum was constructed of practitioners who had a high level of knowledge as to the operating practices carried out within Merseyside Police and collected feedback from their BCU, which they brought with them to implementation meetings that were held on a monthly basis.

To gauge opinions both inside and outside the organisation a mechanism needed to be established for recording levels of confidence. Inside Merseyside Police the levels of attendance at training courses has historically been a very good indicator as to staff's feelings towards the implementation of a new system. With this in mind the decision was taken to monitor the levels of staff attendance at training courses for each BCU. A process of collecting customer feedback was instigated for monitoring opinions of the public and whether the implementation had had any effect on their views of Merseyside Police. The decision was taken by Chief Officers to randomly contact victims of burglary, auto crime, and violent crime to see what their opinions on the level of service provided were. This process was to be conducted prior to the implementation of the generic framework and be carried out on a monthly basis for each BCU. This would prove a useful test of the implementation as it would highlight whether the front line policing had been affected positively and whether the provision of information, which was one of the main problems being experienced, had been improved. Therefore, from a qualitative perspective the following information would be collected for the purposes of evaluation:

- Feedback from BCU staff;
- Training course returns; and
- Public confidence and satisfaction surveys.

Again, as with the quantitative information collected, the qualitative assessment of the implementation was to be performed at a BCU level.

### **9.3.3 Reporting on the Experimentation**

With the information gathered on a monthly basis a reporting mechanism was required to ensure that everyone concerned with the implementation was aware of the successes and failures as they arose. The Chief Officers of Merseyside Police requested that they be informed of progress on a regular basis. As a result each month a presentation would be made to the Chief Officers in relation to information gathered on the subject of the implementation. The process of updating Chief Officers on a monthly basis ensured that the findings from the reports could be filtered down through the organisation so that all staff were aware of current performance.

Papers and presentation made to Chief Officers in relation to the benchmarking of performance can be found in the Appendix of this thesis. As the process of evaluating the implementation developed Chief Officers requested that different information be presented to them, which can be seen in the Appendix. However, for the purposes of consistency within this study only the criteria highlighted in this chapter pertaining to quantitative and qualitative information will be evaluated.

## **9.4 Practicalities: Implementation of a New Crime Recording System**

At the time the implementation of the generic framework was taking place Merseyside Police decided that it needed to introduce a new crime recording system. It was understood that the implementation of the generic framework was underway and it was deemed appropriate to include the introduction of a new crime recording system as part of that implementation. The reason this decision was taken was because it was felt that including the introduction of a new crime recording system could assist with finding a suitable resolution to the problem situation the organisation was experiencing [186].

As has been discussed in this thesis one of the main problems being experienced was that there were too many stand alone systems in place within Merseyside Police. Applying the generic framework to the problem situation quickly



highlighted the fact that the number of stand alone systems should be reduced, starting with the crime recording system. The recommendation made was that Merseyside Police should consider using one of its current systems as the new crime recording system so as to allow the recording of more than one area of policing information in the same place. Chief Officers approved this recommendation in early 2007 with the current custody and court case preparation system (called Niche RMS) being the one selected to be the new crime recording system.

The application of the generic framework suggested that this system be used to ensure that a consistent chain of information could be achieved from the point an officer conducts an arrest through to preparing court files for the Crown Prosecution Service (CPS). This approach was adopted as it would enable a cultural shift to take place within the organisation due to end users being able to understand and appreciate the benefits of recording such information in one central repository.

Users had substantial experience with Niche RMS and using a system already owned by the organisation ensured that minimal funding was required for the implementation process. As a result Niche RMS was chosen as the new crime recording system for Merseyside Police.

However, to ensure that the implementation was conducted properly a mechanism for its introduction needed to be identified. The generic framework identified that a staggered implementation should take place across the six Force BCUs. This would enable the organisation to cope with training demands and also establish a strong platform by which the generic framework itself could be tested. Through staggering the implementation effects of the application of the generic framework, and the new information system, could be very closely monitored against BCUs that did not have Niche RMS for crime recording yet.

To ensure consistency across the Force the staggered implementation took place in three phases, with each phase incorporating roughly the same number of staff.

The phased implementation in 2007 is detailed below:

- Phase one (4<sup>th</sup> June):
  - B – Sefton
  - C – Knowlsey
  - F – Liverpool South
- Phase two (6<sup>th</sup> August):
  - E – Liverpool North
  - Centralised departments (e.g. Counter Terrorism and Internal Security Bureau)
- Phase three (7<sup>th</sup> November):
  - A – Wirral
  - D – St Helens

During each phase of the implementation a team of IT and Police Trainers was made available to BCU staff for a period of two weeks to help staff through the transition from one approach to another. It was deemed appropriate that, depending on positive feedback being received, the support mechanism utilised was consistent to ensure that any views taken from staff could be assessed against that received from different BCUs. Failure to enact a suitable support mechanism could undermine the implementation and prevent accurate evaluations being conducted.

## **9.5 Developments**

As the implementation of the generic framework and new information system progressed it became clear that staff within Merseyside Police was requesting additional work [187].

Firstly staff requested that the Force data warehouse been further developed. This request would not have become possible if users had not increased their confidence in using Force systems, and the request was seen as being a sign that the cultural issue within Merseyside Police was being dealt with. The request was taken into consideration with developments to the Force data warehouse being made with users being fully consulted. To monitor the success or failure of the new functionality in the Force data warehouse feedback was sought during the monthly implementation



meetings. The feedback received was recorded and used to make any further necessary alterations.

Secondly the staff made a request that they be issued with GPRS enabled Blackberries. Again, this represented a large shift in attitudes towards the provision of information as staff felt that they would be willing to collect their own information. The request was made so that front line officer could collect relevant information whilst out patrolling, rather than having to perform this duty in the police station. This allowed officers to spend more time on the streets which led to an upturn in performance, and which in turn also saw an increase in public confidence and satisfaction. To monitor the successes and failures of this development Force performance was closely monitored and included in reports to Chief Officers. To ensure that the increased IT related demand of GPRS enabled Blackberries was tracked system performance was incorporated in the benchmarking reports delivered to Chief Officers on a monthly basis.

It should be noted that the developments were requested during the initial implementation phase of the generic framework and new information system. It was deemed appropriate to continue with the initial implementation first to ensure consistency when testing and to allow suitable time for staff to adjust to the new concepts.

After the third and final phase of the implementation had taken place the process of introducing the two newly requested pieces of functionality took place. Information relating to this second iteration of the generic framework was closely monitored and reported on to Chief Officers. An evaluation of the developments from the second iteration of the generic framework is discussed in Chapter 10.

## **9.6 Chapter Summary**

This chapter has aimed at introducing the processes followed in relation to establishing a suitable testing platform for both the generic framework and the new information system within Merseyside Police.

It is imperative that mechanisms for assessing such an implementation are defined prior to the initial application of the generic framework. This approach ensures consistency and allows an accurate process for testing to be conducted. Failure to enact an appropriate testing platform, or to establish a suitable benchmarking position, corrupts the information gathered and threatens to undermine the results being evaluated.

In Chapter 10 a detailed evaluation of the implementation based on the established testing platform is conducted.



## **10. Evaluation of Experimentation**

### **10.1 Introduction to the Evaluation of Experimentation**

This chapter aims to assess the success or otherwise of the generic framework during its implementation within the case study. The case study was conducted over a period of nine months to ensure that the generic framework had sufficient time to be fully integrated. It is not possible to change people's beliefs and organisational culture quickly, which is why such a long period was established for the implementation of the generic framework.

A detailed evaluation of the generic framework is discussed in the following sections of this chapter.

### **10.2 The New Information System at Merseyside Police – Initial Evaluation**

The implementation of the generic framework started by trying to establish a structured definition of the problem situation. This involved utilising the SSM functionality of the generic framework. Analysis conducted showed that the approach adopted to introduce new systems itself was causing problems. Staff felt that their views had not been taken into consideration when previous systems had been introduced, which was cited as the main reason why the provision of information was not what it could be. The generic framework encourages the analyst working with the current problem situation to develop a conceptual model to determine the most appropriate way of finding and implementing a suitable resolution. The conceptual model below was developed for the case study being discussed.

To ensure that the new interface between the Force data warehouse and the new crime recording system provided information effectively a process of consultation was established. This process was introduced to engage staff in discussions in relation to the interface but also encouraged the staff to feel that they

played an important role in the work that was being carried out. The consultations also produced a very robust set of user requirements and proved to be a very successful research approach.

With staff feeling part of the process it opened up channels of communication by which any assistance or clarification on certain issues could be achieved. This played an important role when the process of testing the interface started. Previous systems had been introduced with the provision of information being dealt with once that system had gone live. Analyses conducted using the generic framework showed that it would be beneficial to test the interface prior to the system go live date by populating a test database with 'dummy' records. This would afford additional time to correct flaws with the interface without affecting operational performance and would ensure that staff involved with the consultation process was happy with the potential provision of information. Should any changes in recording practices emerge at this point they could be taken into consideration without any negative impact to the Force.

The interface that was developed was considered by staff to meet their needs and, once the new crime recording system had gone live, provided information to support operational decisions that had previously been unavailable.

Having utilised the functionality of SSM it was appropriate to VSM's capabilities to ensure that the interface was used and monitored correctly. Step 4 of the generic framework links SSM and VSM and enables the switch in viewpoint from defining the problem situation to implementing an appropriate solution.

The outcome of the switch in viewpoint between SSM and VSM ensured that robust and clearly defined interface specifications were developed. However, the interface needed to be constructed by the right people at the right time. At this point in proceedings the R<sup>5</sup> model, from Step 5 in the generic framework, was implemented. Due to the confidential nature of the information contained within the R<sup>5</sup> analysis conducted for this UK Force the results cannot be displayed. However, the analysis ensured that the right resources were provided for the development of the interface, along with establishing which legislative constraints must be adhered to.



The newly developed interface allowed staff to link custody records, victim and offender details, property, court case files and crimes together from one central repository. It had been possible to gather such information before, however, the linkages that connected all this information together had not previously been in existence. This innovation enabled the Force to perform crime analysis much quicker than it had been possible previously. This has led to enhanced decision making at management level, which, in turn has seen a positive effect on Force performance. Within the first few weeks of the implementation of the new crime recording system and its interface crime levels had dropped across the Force. Three months after the implementation of the new interface crime levels across the whole Force had dropped by 11% (and 23% after six months) with detections showing a rise of nearly 6% [188, 189, 190]. Whilst it is still considered premature to announce the success or otherwise of the case study as a whole it is clear to see that the early signs are positive.

In addition to being able to link key information throughout the Criminal Justice process it is now possible for the Force conduct analysis into how many suspects have been named recently, enabling management to deploy officers to carry out the necessary arrests. The information provided also highlights to the management ranks how the suspect was named, including whether they have been named through DNA hits or through fingerprint, footwear and forensic matches. This information has proved invaluable when incorporated into staff deployment plans and crime investigation.

Throughout the investigative process each officer's involvement can now be tracked. The ability to monitor which officer is in charge of particular cases, who carried out various arrests and who have conducted interviews is new to the Force. In fact, this is new to UK Police Forces. Throughout the life of a custody record, crime record or court case file each officer, and their link in the process, can be tracked. This represents an innovation in respect of auditing and staff deployment.

Briefing tools have also been included in the implementation of the new crime recording system and its interface. Management level staff can now produce detailed reports on crime that has happened in their area for any given period, which includes

details of victims, suspects, offenders and property involved at the click of a button. These briefings are now produced each morning and are used when determining staff deployments or operational planning. In addition to this, information is highlighted to officers when carrying out their day-to-day duties. For example, if an officer has conducted an arrest they will be informed if the person arrested is wanted on warrant, has missed their bail, or is a named suspect for another crime.

As a result of the newly developed interface the Force is considered to be ahead of most in the UK through its use of GIS, with a vast increase in the provision of information being experienced. GIS now forms the backbone of information used within the Force and is used to support operational planning and key strategic decision making.

GIS has been quickly picked up on by other UK Forces for their use in a variety of different operational situations though their use has not been fully utilised for crime mapping or analysis. There has recently been a flood of publicity as Forces both here and in the United States have invested in GIS. However these systems have been mainly used for mapping incident data, and for providing maps for describing scenes of incidents to a court. Incidents from the emergency system can be displayed on screen so that the operator can allocate the most efficient resource to deal with the particular event.

To complete the consultation process a series of presentations and workshops were carried out to communicate the introduction of the new interface to staff that would be using it to collect information. The presentations and workshops were received well by staff with Bluestar, the company that owns the Force data warehouse, reporting a sharp upturn in the number of crime related searches being carried out.

Accurately guiding responding officers straight to the scene of the incident can save valuable time and these same principles have also been applied to other emergency services.



With incident deployment improving the next progression looks at providing officers with the information they need to carry out their front line duties. The future possible application of GIS for crime pattern analysis as a tool to assist officers seems the next step in solving the information starvation problem. The Force's ability to conduct in depth mapping analysis has now been greatly enhanced due to the additional provision of geocoded information. Staff can now assess details of all linked persons or crimes on dynamic maps that run through the newly developed interface.

Given the approach adopted through utilising SSM's functionality staff are actively seeking new ways to collect the information they require. The culture in the organisation has seen a huge shift whereby staff feel confident in their ability to collect accurate and relevant information, and are willing to conduct their own analysis too. A direct result of this cultural change has seen an increase in performance. In respect of a UK Force this represents a fall in crime levels and an increase in detection rates. This work will continue to develop in the coming months and years as the information provided through mapping has become more powerful and assists the investigative process better than it has done before.

Following on from the implementation of the new interface within the Force requests are being made to further enhance it by incorporating new technology. The introduction of GPRS enabled blackberries for dynamically briefing officers whilst out on patrol is not too far away. This will provide detailed information to officers through tracking their location and determining what needs to be brought to their attention whilst they are on patrol.

The introduction of dynamic resource tracking is in the early stages of being developed. Through dynamically tracking officer movements in real time deployment decisions can be taken in response to situations that may arise. This will be a powerful new development in the world of policing and is only possible due to the implementation of the new interface between the Force data warehouse and the new crime recording system.

The generic framework in this case study highlighted that the problem that existed within the Force had arisen from poor communication when new systems were being introduced. By establishing effective lines of communication and understanding the cultural beliefs of staff within the Force a new interface was developed that would improve the provision of information. The technological capacity to develop the interface was already in existence within the Force. However, changing the processes that led to the development of the new interface ensured that improvements to the provision of information were achieved.

### **10.3 Methodological Approach**

The following section of this chapter aims to establish the appropriateness, objectives and success of the development and implementation of the generic framework. Having conducted an initial evaluation of the case study within Merseyside Police it is now appropriate to perform a more detailed analysis of the generic framework.

#### **10.3.1 Appropriateness**

This evaluation required that the generic framework be implemented in a real world situation, which is covered in the case study. To conform with the concept of developing an approach that was able to deal with the 'whats' and the 'hows' of problem situations the generic framework required functionality from both hard and soft methodologies. In addition to this the methodologies chosen would also be required to have self-adaptive functionality. This would greatly assist staff working in real world situations as a degree of self-adaptation and autonomy would help restore a healthy equilibrium.

The literature review establishes a baseline for the generic framework requirements. This, however, also provides a set of prerequisites for the methodologies that were to be taken into consideration for the generic framework.



Soft Systems Methodology was deemed the most appropriate of the soft approaches due to its extensive application to complex problem situations. In particular Soft Systems Methodology has been applied to situations which require organisation restructuring and changing the cultural beliefs of staff that work within it.

Viable Systems Model was selected as being the most appropriate hard methodology due to its ability to cope with changes in its operating environment and its inherent self-adaptive functionality. The benefit of choosing the two methodologies mentioned above for the development of the generic framework is that they are both flexible enough, with overlapping functionality, to ensure that they can be integrated fully without compromising their core abilities.

With the development and subsequent implementation of the generic framework it is appropriate to perform evaluation to monitor which areas were successful and where potential improvements could be made.

### **10.3.2 Objectives**

For the purpose of evaluating the generic framework a case study was conducted. Whilst the case study was being carried out each step of the generic framework was being monitored for its compliance with both its requirements and adherence to resolving the problem situation. This chapter aims to assess the success or otherwise of the generic framework through both qualitative and quantitative information gathered.

Stability is seen as being crucial to the success of the generic framework. Stability is achieved if the generic framework is implemented within the bounds of its control rules and if it can sustain itself as a stand alone entity.

Self-adaptation monitors how successfully the generic framework is able to comply with its self-adaptive and autonomic roots without requiring significant input from practitioners when changes in its operating environment are experienced. This establishes how robust the generic framework is in addition to its self-adaptive capabilities.

Performance measures of the generic framework were conducted through management information collected from the organisation involved with the case study. Should organisational performance be impacted upon negatively the generic framework will be expected to implement a resolution strategy. If positive performance is experienced the generic framework will be expected to assess approaches to ensure that current performance levels are maintained.

### **10.3.3 Approach**

The process of evaluating the generic framework involves monitoring performance of the case study organisation. This is not to monitor the organisation itself, however, this acts as a suitable measure for the effectiveness of the generic framework.

Given that the application of the generic framework is meant to facilitate the possible changing of culture within an organisation it is appropriate to allow several months for the case study to be carried out. By conducting the evaluation on too short a timescale the results of the implementation of the generic framework could become corrupted and misrepresent its application. Within Merseyside Police a period of nine months was established for the case study as this would allow sufficient time for the generic framework to be implemented.

Through implementing the generic framework in a variety of different environments within Merseyside Police it will be possible to assess its ability to adapt to change and monitor its tolerance levels before control resolution strategies were introduced.

### **10.3.4 Requirement for Evaluation**

Before the evaluation can commence it is imperative to identify the problem situation that the generic framework is being applied to. The success or otherwise of the generic framework can only be monitored against the problem situation it is being



implemented to counteract. Therefore, once the process of defining the problem situation has been completed the relevant boundaries for the evaluation can be established.

The intention for the boundaries is to determine when conflicts should be detected and whether resolution strategies are to be implemented. Once one of the boundaries has been exceeded and an appropriate measure has not been implemented this would suggest there has been a failure in the generic framework.

### **10.3.5 User Involvement**

This chapter assesses the generic framework based upon experience collected from the case study within Merseyside Police. This required fully implementing the generic framework within the organisation and ensuring that users are involved in this process.

As the initial evaluation of the generic framework shows this took the form of a newly developed information system within Merseyside Police. This approach was adopted as it enabled all staff across the organisation to utilise the new functionality offered by the application of the generic framework. This not only provided a new way of implementing a resolution to the problem situation but also raised awareness in the work that was being carried out.

### **10.3.6 Environment**

Due to the sensitive nature of the work Merseyside Police conducts it was necessary to develop the new information system in a testing environment. Once the information system had been developed it was then introduced to the organisation in a 'live' environment.

In some circumstances it is appropriate to develop such information systems in a live environment, however, the application of the generic framework acknowledged

constraints that needed to be imposed on access to information before the developmental stage had been completed.

#### **10.4 Quantitative Evaluation of the Generic Framework**

Merseyside Police is requested to provide information to the Home Office to ensure that its performance is monitored. There are some areas whereby the organisation has not been able to meet these requirements, which is something the problem situation looked to try and resolve as part of its implementation.

Due to the nature of the modern policing environment performance related management information is provided on a daily basis within Merseyside Police. Changes to current processes and approaches can be monitored through the management information provided. In addition to this the newly developed functionality offered to staff through the new information system enhances the provision of information so that requirements can be tailored to meet specific demands. For example, users can retrieve information in relation to various operations, although the nature of the generic framework will determine that the information system will make certain decisions on the user's behalf in relation to what can be presented.

As discussed in the case study experimentation of the generic framework the transfer of one crime system to another took place in Merseyside Police in 2007. Experience has shown that during such times organisational performance can be negatively affected as staff struggle to come to terms with the new technology or process. The generic framework identified that recording crime information on the same system as an existing Force system would be beneficial. This would allow information from different roles within the organisation to be recorded in one place and linked together. This accommodated user concerns about the introduction of another system as the new crime recording system was a tool they were already used to using. This catered for the cultural beliefs and requirements of staff within Merseyside Police, which will be covered in the qualitative evaluation in this chapter. However, this approach also catered for the quantitative needs of the organisation as it



would provide a solution whereby information for several key organisational processes could be recorded in the same place. In the case study this meant that custody, crime and court case preparation information could all be recorded, and therefore reported on, together as a move towards an integrated criminal recording system was first introduced to the organisation.

The response towards this approach initially was positive as the benefits to Force performance could be seen by staff. During the phased implementation the initial feelings towards the approach were maintained as management information collected showed a positive effect on performance. In fact, as will be discussed in the following section of this chapter, staff in parts of the Force that had not yet ‘gone live’ with Niche for crime recording were looking forward to its implementation.

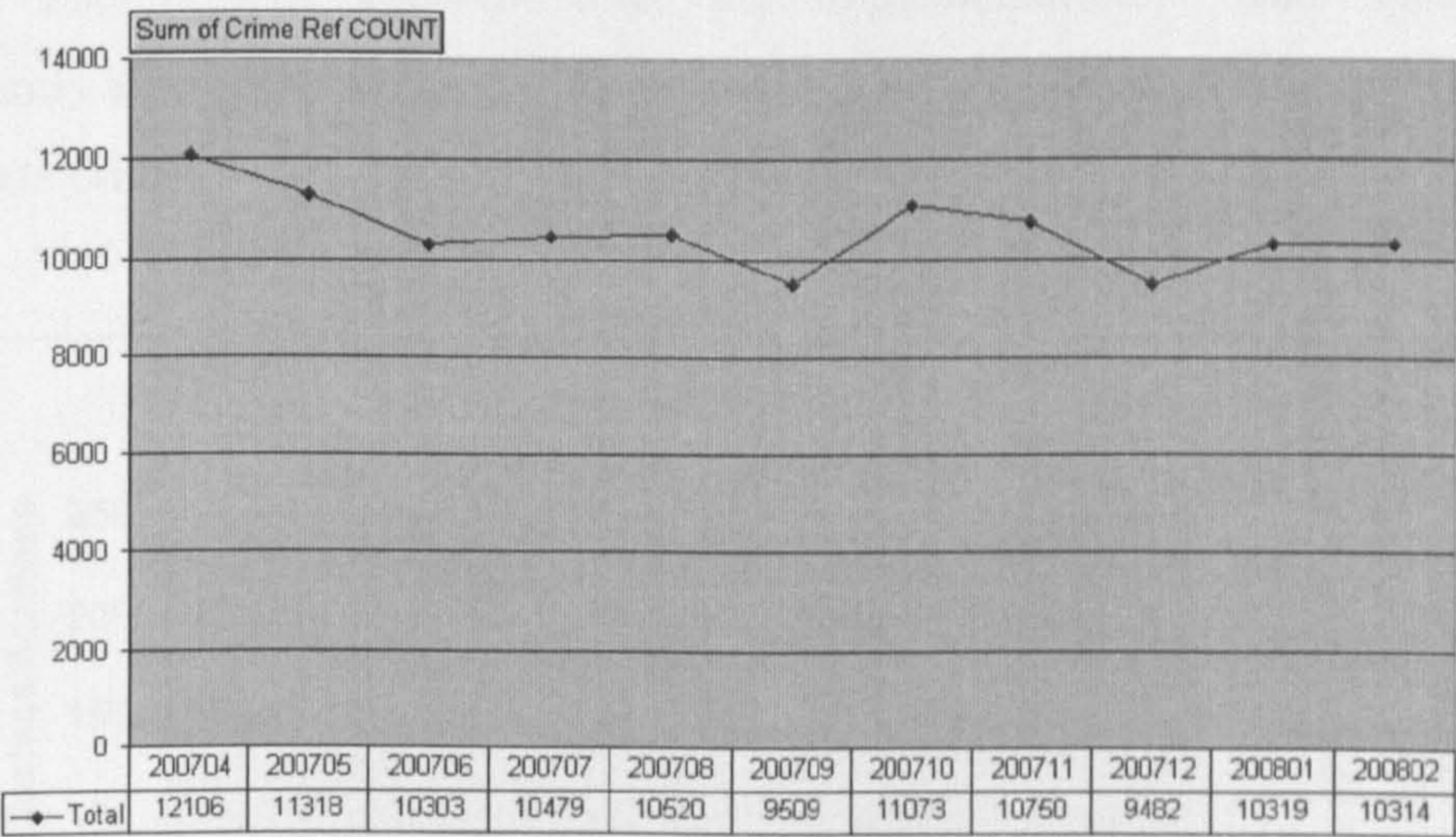


Chart 1: Chart demonstrating the positive effect the implementation of the new approach within Merseyside Police has had on the level of crime.

Although the implementation of the new approach had an effect on the crime rate within the Merseyside area there were also many other benefits experienced. The chart below demonstrates the average custody airlock times across the Force (an airlock time being the time between arriving at a custody suite and being booked into a cell).



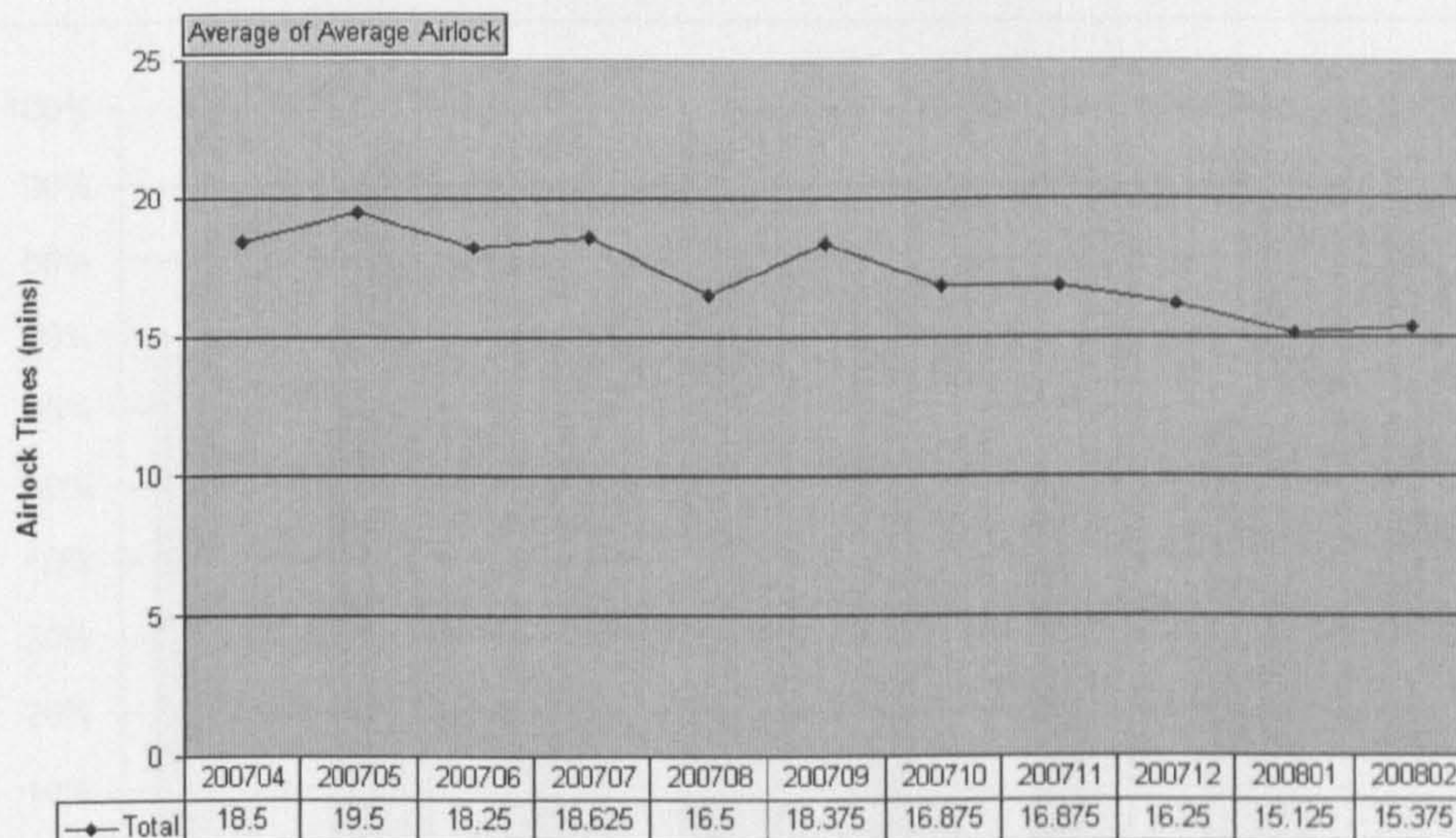


Chart 2: Chart demonstrating the drop in average custody airlock times across the Force.

In addition to the above noticeable improvements to performance other benefits also became apparent after the implementation. Data quality was significantly enhanced, which led to an improvement in the organisation's ability to investigate crime.

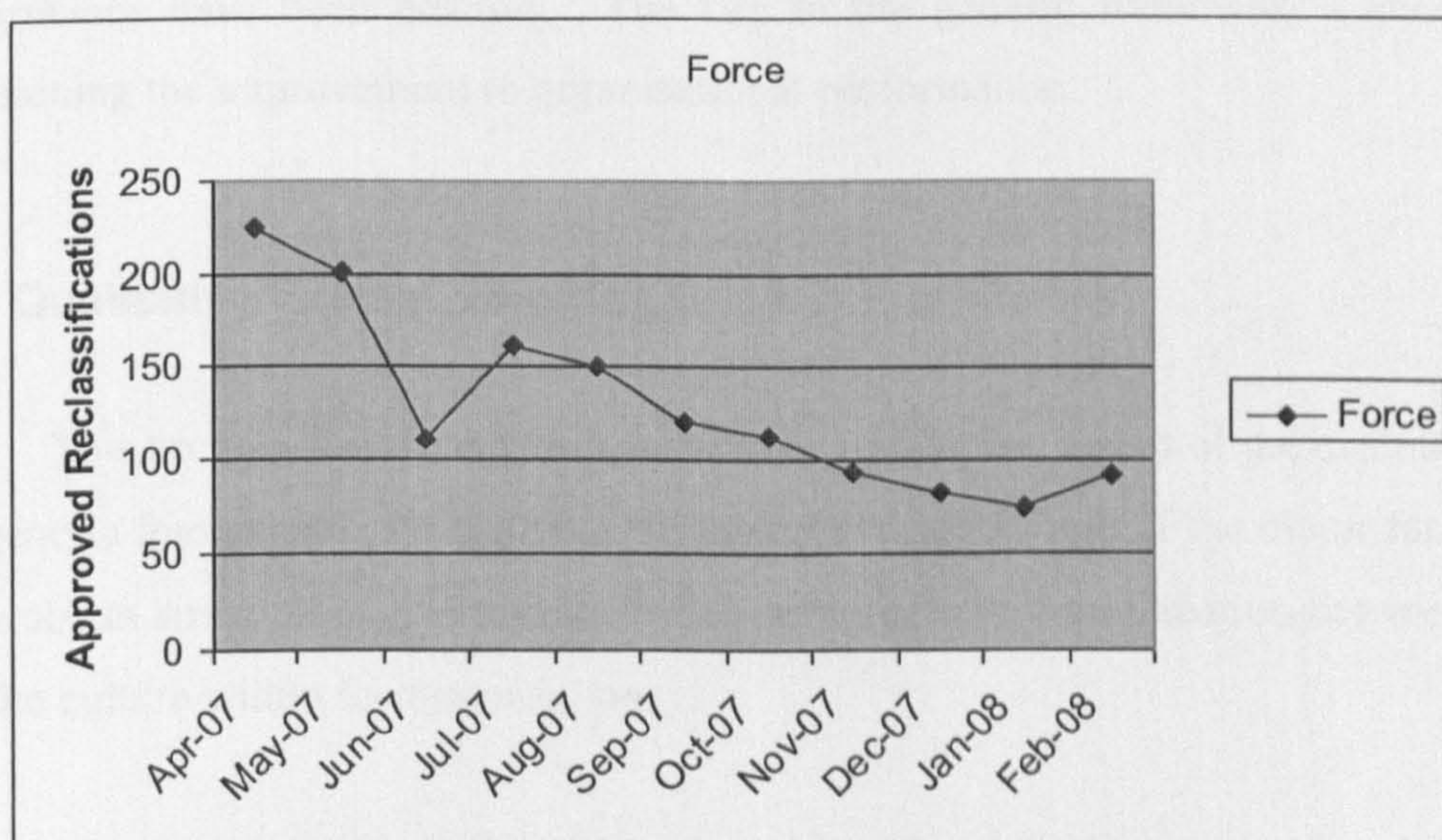


Chart 3: Chart demonstrating the fall in the number of crimes that require reclassification. This highlights the fact that more crimes are being recorded under the correct initial classification.



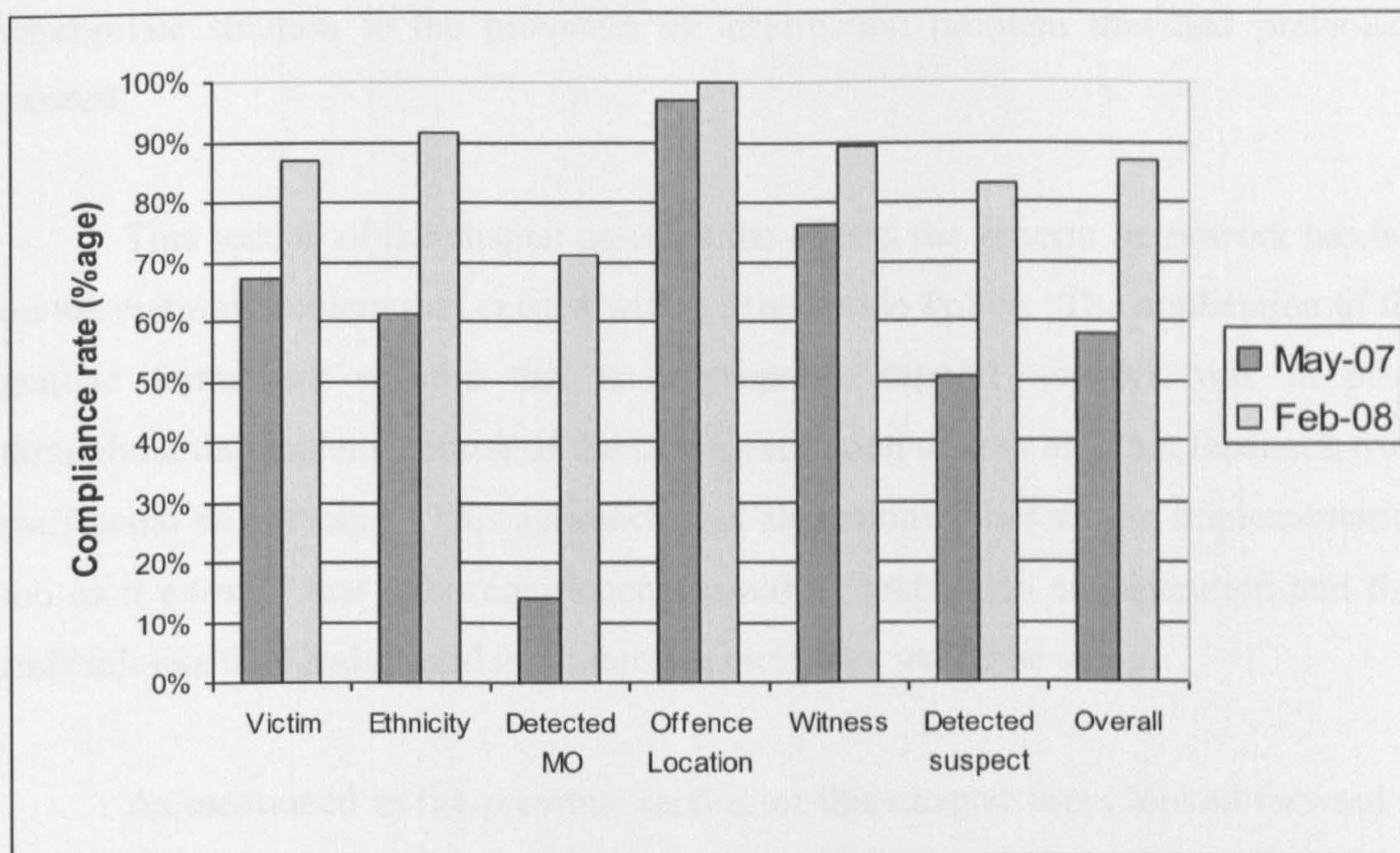


Chart 4: Chart demonstrating results of a data quality audit on crime records. The new approach adopted by Merseyside Police shows a sharp rise in compliance to data quality standards, which significantly enhances the investigative process.

As can be seen in the information provided above the effects on Force performance have been positive. The key to the generic framework's success is maintaining the improvement to organisational performance.

## 10.5 Qualitative Evaluation of the Generic Framework

This section of the chapter discusses the qualitative aspect of the evaluation of the generic framework. As has been discussed previously one of the major factors in the problem situation of Merseyside Police was that of communication between staff and the culture within the organisation.

The implementation of the generic framework was conducted in such a way so as to find an appropriate resolution to this problem in addition to attempting to improve Force performance. The previous section of this chapter has shown that changing the IT systems for crime recording and the development of a new information system greatly enhanced the provision of information throughout the organisation. This led to improved Force performance and can be seen as an



appropriate solution to the provision of information problem that had previously existed.

This section of the chapter assesses the effects the generic framework has had on the cultural problem that existed within Merseyside Police. The application of the generic framework ensured that an appropriate support network was in place throughout the implementation of the new information system and that feedback from staff could be gathered. This approach was also adopted before the implementation too as it ensured that user requirements could be taken into consideration and that staff felt like they had a valid input into the tools they would be using.

As mentioned in the previous section of this chapter users looked forward to the implementation of Niche for crime recording in their part of the Force and training returns showed that attendance at non compulsory courses were over 80%. The breakdown for each of the three phases of the implementation is detailed below (it should be noted that attendances increased as the implementation progressed as staff culture started to change to embrace the new approach):

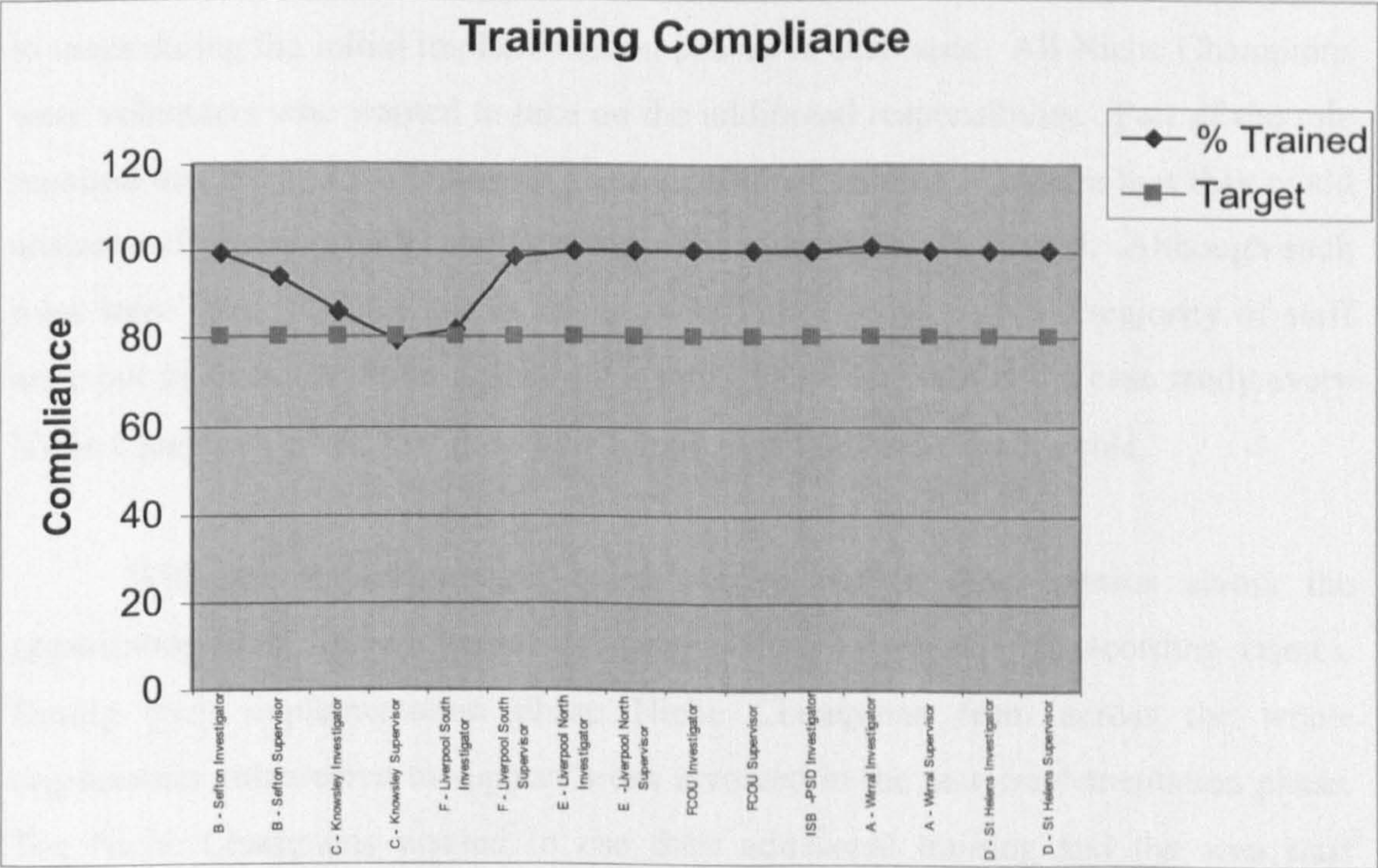


Chart 5: Chart demonstrating training compliance throughout the phased implementation of the new approach within Merseyside Police.



Traditionally the organisation has struggled to achieve its target of training 80% of staff prior to go live date, however, this time 100% compliance was experienced in almost every part of the Force.

To ensure that appropriate qualitative information was gathered for evaluation purposes a mechanism for communication was introduced. This consisted of establishing a web based user forum where staff across the organisation could discuss any issues or difficulties they were experiencing, share best practice with other staff, or to highlight areas for potential improvement. Previously in Merseyside Police web based forums had been implemented, however, they had not proved to be successful as staff tended not to use them. During this case study though the user forum proved popular with staff of all ranks and backgrounds sharing ideas and communicating their views. Feedback and appropriate responses from the forum can be found in the Appendix of this thesis.

In addition to this each area of the Force had designated 'Niche Champions' who took up responsibility for communicating new developments to staff and collected user feedback. The role of the Niche Champion was to also provide support to users during the initial implementation phases in each area. All Niche Champions were volunteers who wanted to take on the additional responsibility. Part of the role required that Niche Champions be given additional training to ensure that they could answer staff queries, with training returns showing 100% attendance. Although such roles have been implemented in Merseyside Police previously the majority of staff were put in such a position against their will. However, during the case study every Niche Champion stated that they were happy to volunteer for such a role.

With the implementation being carried out in three phases across the organisation staff found themselves using different systems for recording crimes. During each implementation phase Niche Champions from across the whole organisation volunteered to support areas involved in the new implementation phase. The Niche Champions wanted to use their additional training and the area staff benefited from the support offered.

Sergeant Alan Harrison stated at a user group meeting that:

*“I have never been involved in such a pain free implementation. Initially I had some feelings of trepidation when it was announced the crime system would be changing again. However, I heard that the phase one implementation went really well and I was keen to go on the training to see what all the fuss was about. I have to say that the way we do things now makes my job much easier than it used to be.”<sup>1</sup>*

Although the implementation of the generic framework had been received well amongst staff it was still imperative that the changes in organisational culture being experienced had an effective end result in the service offered to the public. The trial implementation of GPRS enabled blackberries as part of the second iteration of the generic framework proved to have positive results. At the time of writing this thesis the second iteration of the generic framework is still in progress. However, with staff feeling much more confident in their ability to retrieve accurate and relevant information the introduction of GPRS enabled blackberries was met with an encouraging response. During an initial testing phase Sergeant Jason Lane commented:

*“Now I don’t even have to be in the station to collect the information I need. Constables in my block haven’t used their radios too much since we got the blackberries. It is much quicker to get the information ourselves and we can even collect mugshots of people we’re stop searching on the street through our blackberries. News of this seems to be spreading as we have noticed that we don’t see so many groups of youths congregating in parks and so on any more.”<sup>2</sup>*

With the additional information available through the new information system Assistant Chief Constable Helen King requested that a new mechanism of monitoring the way in which victims of crime were treated be instigated. This scheme was introduced in late 2007 and assessed the victim’s opinion of what they thought of the way Merseyside Police handled their particular incident. The results showed an

---

<sup>1</sup> Niche user group meeting held at Southport police station on 6<sup>th</sup> September 2007.

<sup>2</sup> Niche user group meeting held at Smithdown Lane police station on 14<sup>th</sup> March 2008.



increase in public confidence and satisfaction, which represented an improvement in data quality and a change in organisational culture. The results are detailed below:

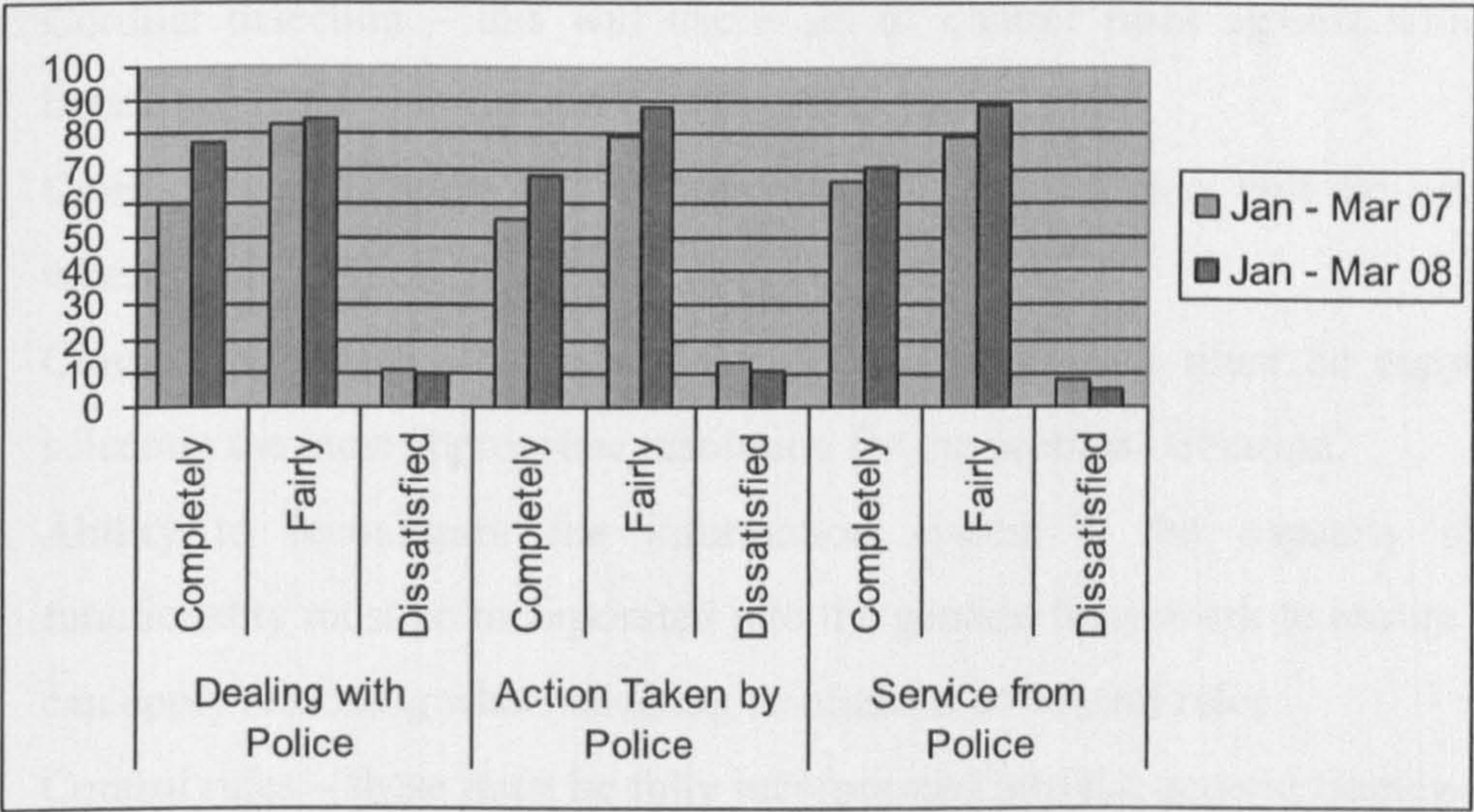


Chart 6: Chart demonstrating the rise in percentages of public confidence and satisfaction through surveys conducted with victims of crime.

The increase in public confidence and satisfaction shows that a cultural shift in Merseyside Police has taken effect. This also highlights that some of the changes that have taken place within the organisation have been passed on to the public through an improved service.

### 10.6 Discussion

It is a significant challenge to produce conclusive evidence of the benefits, merits, effectiveness, correctness, appropriateness and completeness of such a generic framework. That said, the quantitative and qualitative evaluations provided in this chapter do suggest that the implementation of the generic framework has had a positive effect on finding a suitable resolution to the problem situation within Merseyside Police.



The requirements of the generic framework that were initially established were as follows:

- Conflict detection – this will use a set of control rules against which the behaviour of the information system can be monitored.
- Conflict identification and classification – this function will be activated whenever a conflict or failure has been detected.
- Conflict resolution strategies – the generic framework must be capable of selecting the most appropriate resolution for the problem situation.
- Ability to reconfigure the information system – the capacity of this functionality must be incorporated into the generic framework to ensure that it can apply reasoning when selecting resolutions or control rules.
- Control rules – these must be fully incorporated into the generic framework so that the information system can examine its internal and external process and activities.

This thesis discusses how the generic framework was constructed with these five core requirements present throughout the developmental phase. This section of the chapter will now assess whether the implementation of the generic framework met the five core requirements as outlined above and discussed in Chapter 5.

### **10.6.1 Conflict Detection**

The application of the generic framework highlighted the necessity for change within Merseyside Police. The initial phase of implementation of the generic framework pointed to difficulties that were affecting performance in the organisation and that not all of them could be solved through introducing a new IT system as a means to a solution.

During the implementation of the generic framework the conflict detection capacity highlighted areas for potential improvement and future possible conflicts. Indeed, it was this functionality of the generic framework that led to the eventual introduction of GPRS enabled blackberries during the second iteration.



### **10.6.2 Conflict Identification and Classification**

The application of the generic framework highlighted that the problems being experienced in Merseyside Police were due to information starvation and organisational culture. If the generic framework had not contained such functionality it would not have been possible to implement an appropriate solution. As with the conflict detection capacity an appropriate resolution could not have been established had the generic framework not contained this functionality.

During the implementation of the generic framework the new information system was developed as a result of identifying and classifying the information starvation and cultural problems being experienced.

### **10.6.3 Conflict Resolution Strategies**

The conflict resolution strategies functionality played a large part in the design of the generic framework and the new information system. As with the previous two items of functionality the generic framework would not have been able to introduce the necessary changes without having the capacity to develop robust conflict resolution strategies.

In the case of Merseyside Police the application of the  $R^5$  model led to the development of a flexible but robust information system that resolved the major factors in the problem situation that existed. During the second iteration of the generic framework it was this functionality that facilitated the introduction of GPRS blackberries and alterations to the new information system based on user feedback.

### **10.6.4 Ability to Reconfigure the Information System**

As has been discussed in the previous sections of this chapter the generic framework needed to be able to reconfigure the information system to meet the needs of Merseyside Police. In the case of all large organisations this functionality is crucial as requirements can change on a daily, and in some cases hourly, basis. In the modern

policing environment this can be due to operational demand or through changing legislation.

During the case study it was necessary to start a second iteration of the generic framework owing to increased reliance on the information system and the desire to utilise its functionality on the streets with front line officers. As discussed above this primarily took the shape of GPRS enabled blackberries but also enhancements were made to the information system through make different strands of information available that had not previously been available to the organisation. The culture within Merseyside Police seemingly took its biggest changes during the second iteration of the generic framework as staff could see that this approach was flexible and able to adapt with them. This ensured that staff no longer felt resistant to change and that they would be supported whilst working in an ever changing environment.

#### **10.6.5 Control Rules**

As with the conflict resolution strategies functionality the generic framework was able to apply the R<sup>5</sup> model to develop resilient control rules that could cope with change but also ensure system integrity.

Throughout the implementation of the generic framework within Merseyside Police the control rules ensured that nobody was able to retrieve information that they should not have been permitted to view but also that they were able to override certain decisions that the new information system had made on their behalf. In a modern policing environment staff can move around the Force several times a year and their information needs can change frequently. It would be far too labour intensive to manually change the new information system and so control rules were developed that would allow self-adaptation and autonomy to ensure such changes were managed dynamically.

Should external changes to the operating environment of Merseyside Police be recognised the control rules ensured that these were accommodated. Throughout the implementation of the generic framework the Home Office made legislative changes,



whilst the Force itself introduced new operations, all of which had to be catered for. The generic framework and information system were able cope with these changes and update the necessary policies and directives of the control rules when applicable.

## **10.7 Chapter Summary**

This chapter presents the quantitative and qualitative evaluation results where the main aim has been to try and demonstrate the success or otherwise of the generic framework and the new information system. In meeting this aim we have been able to analyse the relationship between the newly developed functionality of the generic framework and its application in a real world case study. In this case Merseyside Police was the chosen organisation with positive results being demonstrated following on from the implementation of the generic framework and the subsequent development of a new information system.

This chapter also set out to demonstrate whether the generic framework had adhered to its requirements during the implementation with Merseyside Police. It is imperative that the generic framework can be utilised by any organisation and in any context, which would not be possible if it did not adhere to its original requirements. The principles of self-adaptation and autonomy are key to the generic framework and it has been demonstrated in the quantitative and qualitative evaluation that these principles have been maintained consistently throughout. The ability of the generic framework to monitor itself played a significant role in its adherence to its original aims and objectives, whilst ensuring compliance with the five core functionalities that form the backbone of its development.

# **11. Conclusions**

## **11.1 Introduction to Conclusions**

This chapter sets the goal of drawing together conclusions to the work carried out as part of the studies covered in this thesis. The case study was conducted over a period of nine months to ensure that the generic framework had sufficient time to be fully integrated. This chapter discusses the contributions and achievements made in this study along with a thesis summary and an overview of future work to be carried out.

## **11.2 Motivations and Approach Summary**

Over recent years the concepts of self-adaptation and autonomy has seen a vast amount of research interests being pursued. This has mainly concentrated on design, analysis and management related theories and models to support dynamic decisions making in software solutions. The idea of creating a generic framework with these principles taking a core role has not been studied before.

IBM [191] in particular has invested heavily in driving forward autonomic computing to achieve self-managing capabilities for the systems they build. The studies carried out by IBM also expand into the fields of self-healing and self-protecting systems.

Much effort has gone into design, analysis and management related theories and models to support dynamic decisions making in software solutions, however, this thesis covers a research study that looked at integrating such capabilities into an information systems approach. The motivation for such research was to attempt to develop a new approach that would lead to the creation of a new field of thinking, which included the concepts of self-adaptation and autonomy. This would enable such an approach to cope with unexpected changes within its operating environment and to select appropriate resolution strategies without requiring user interaction at



every stage, whilst still ensuring adherence to control rules in an effective manner. To achieve this ambition several challenges have had to be addressed.

Development of a new information systems thinking approach was necessary to develop a new methodology that could support managing a function with self-adaptive and autonomic principles and be able to take appropriate corrective action.

To achieve this the following needed to be taken into consideration:

- Determine suitable management boundaries by detecting, filtering and administering repairs.
- Coordinate conflict resolution strategies.
- Support dynamic configuration and reconfiguration to enable the control functionality of the generic framework to adapt repair and resolution strategies according to changing requirements and environments.
- Support functionality that can determine policies and control rules to assist decision making.

A suitable case study was necessary to implement the new approach, and a new information system, to a real life problem situation and monitor its capacity to monitor, repair and reconfigure itself taking into account operating restraints and cultural issues.

This thesis has provided details of the development of a new generic framework and its inherent functionality. In addition, this thesis has discussed the integration of self-adaptive and autonomic functionality whilst overcoming the challenges detailed above.

The generic framework developed in this research study is based upon and understanding and appreciation of existing knowledge in this field. To assist the development of the generic framework, along with the selection of appropriate methodological approaches, the research carried out assessed the key fields of related interest discussed below.

Self-adaptive systems research was conducted into feedback and feedforward mechanisms to provide the capability for continuous evaluation of system behaviour and the ability to select appropriate resolution strategies.

Autonomic computing concepts research was carried out in relation to the theoretical side of the autonomic application of control rules and repair and resolution strategies. This research also evaluated the merits of supporting, and including the functionality of, communication channels that facilitate the ongoing management of a system in an ever changing operating environment.

Extensive research into the field of information systems thinking was conducted to ensure appropriate functionality was integrated into the newly developed generic framework. This particularly covered current methodological approaches and the hard and soft dichotomy.

It is worth noting that this study has provided a theoretical and practical application of the concepts put forward by this research. Management capability within the generic framework is responsible for the control and system monitoring, along with conflict detection and the selection of suitable resolution strategies.

Adaptive capacity allows the generic framework to control, coordinate and reconfigure itself to ensure that performance is fully maintained in changing operating environments.

Communication channels within the generic framework has the ability to learn from experience and update policies and strategies where appropriate. Any changes at this point can then be communicated to interested parties.

Developing a control base from which activities can be repaired when necessary, resolution strategies can be implemented and where operating boundaries can be determined.



A practical representation of the implementation of the generic framework, and subsequent new information system, within Merseyside Police is provided in this thesis.

### **11.3 Contributions**

The main contribution of this study is the development of the generic framework that can be applied to any problem situation due to its ability to cope with changing environments through its use of self-adaptive and autonomic concepts. The generic framework facilitates the ability to self-manage through containing functionality that can monitor, repair and reconfigure should any conflicts, failures or inconsistencies be highlighted in the environment within which it exists.

In addition the study has demonstrated its application to a real world problem situation and highlighted its ability to cope with changing environments. Whilst being able to adapt and respond to situations as they arise the generic framework has the ability to establish control rules and policies, which can be updated and monitored according to need for such functionality to be instigated. The concepts of self-adaptation and autonomy underpin the development of the generic framework in respect of self-management and self-healing, however, the functionality associated with selecting and applying resolutions strategies contains theories put forward by the BDI Model. The application of the BDI Model to information systems thinking is new to the field and allows flexibility within the generic framework when coping with change and selecting appropriate resolutions strategies. The functionality incorporated into the generic framework through the BDI Model is as follows:

- Beliefs – are represented by two structures. Firstly, a model of the external environment and secondly the internal environment that exist within the information system.
- Desires – represent a set of actions and desired aims which need to be achieved at a specific time.

- Intentions – is determined by a process of deliberation, which translates desires with respect to the current beliefs about both the environment and the current weltanschauung.

The application of the  $R^5$  model to information systems thinking is also a significant new contribution.

The introduction of the  $R^5$  model to the generic framework adds the following functionality:

- Resources;
- Rules;
- Responsibilities;
- Regulations; and
- Recommendations

Through each iteration of the generic framework the  $R^5$  model is implemented to ensure that appropriate resolutions are developed. That said a key part of the functionality of the generic framework is that although it leads the user towards potential solutions the ability to manually override decisions being made at any point is always available.

Whilst containing new functionality it should be noted that the approach adopted allows for both hard and soft information systems thinking to be incorporated into one methodology. This is a new development in the field of information systems thinking and provides the analyst with the opportunity to gain a full understanding and appreciation of problem situations from both perspectives.

The generic framework has been developed with three main strands that allow for the full functionality of self-adaptation, autonomy and the capabilities from SSM and VSM to be embraced. These strands are combined, rather than several new strands that operate in isolation, that enable full functionality of self-adaptation, autonomy and the capabilities from SSM and VSM.



The control architecture also contains three combined strands. The first strand contains the registration and discovery functionality of the framework. This establishes a monitoring capability and determines when failures or conflicts have been detected.

The second strand contains the management and control of failures or conflicts that have been detected. This requires that accurate information on the problem situation is gathered and contains self-adaptive and autonomic capabilities.

The third strand contains the repair functionality of the framework. This strand contains self-adaptive and autonomic capabilities of the framework and establishes how to repair the detected failures or conflicts.

This enables fully integrated feedforward and feedback loops into the generic framework. This forms a crucial part of the generic framework's monitoring and discovery functionality. The importance of these feedforward and feedback loops is that they start a process that must determine the control inputs, check the information system and its operating environment, classify a detected failure or conflict, establish which conflict resolution is most appropriate and ensure that adaptations and information system reconfigurations run smoothly.

To this end, the control architecture, or mechanism, achieves the aims of self-adaptation and autonomy through a continuous cycle of detection, identification, classification and resolution of failures or conflicts. The feedforward and feedback loops will be utilised to continuously monitor the mechanism itself, which ensures that the generic framework has a fault tolerance capacity.

## **11.4 Achievements**

This research has led to the development of a new generic framework that has been fully applied to a problem situation within a large public sector organisation. This thesis provides details of the requirements for the generic framework to ensure

that it fully meets its aims and objectives in relation to the principles of self-adaptation and autonomy.

The requirements are as follows:

- Conflict detection – this will use a set of control rules against which the behaviour of the information system can be monitored.
- Conflict identification and classification – this function will be activated whenever a conflict or failure has been detected.
- Conflict resolution strategies – the generic framework must be capable of selecting the most appropriate resolution for the problem situation.
- Ability to reconfigure the information system – the capacity of this functionality must be incorporated into the generic framework to ensure that it can apply reasoning when selecting resolutions or control rules.
- Control rules – these must be fully incorporated into the generic framework so that the information system can examine its internal and external process and activities.

The above requirements have been fully incorporated into the generic framework. This is seen as a significant achievement as it combines several strands of functionality that have not been implemented together in the field of information systems thinking previously.

In addition to this the generic framework has been developed to support self-adaptation and autonomy through flexible, accessible system control functionality in order to monitor, repair, coordinate and reconfigure as and when required. This ensures stability and can be conducted without any user interaction. Once defects or inconsistencies have occurred they can be dealt with through the generic framework's functionality whilst storing relevant information, updating policies and control strategies, and providing information to interested parties. This is seen as being not only a significant contribution to the field of information systems thinking but is also a major achievement against the initial aims specified in this thesis.



The generic framework has been developed and its implementation into a real world problem situation has been discussed. The application of the generic framework, and subsequent development of a new information system, is a significant achievement as it provides substantial proof of concept and evidence that it works in complex situations that require its full functionality.

Merseyside Police is a large public sector organisation that must operate within very tight budget constraints. As a result the implementation of the generic framework within Merseyside Police needed to work within these boundaries, along with those imposed by the Home Office, as well as being able to understand the culture that existed in the organisation. Again, this is seen as being a major achievement as Merseyside Police has shown that it fully embraced the concepts put forward in the generic framework, with very positive results demonstrated in this thesis.

## **11.5 Thesis Summary**

This thesis has provided a new approach to the field of information systems thinking through encompassing the principles of self-adaptation, autonomy and linking methodological concepts. A detailed description of the study undertaken is presented in this thesis.

Chapter 1 provided an introduction into the motivations and challenges ahead and how the work was to be undertaken.

Chapter 2 assesses methodologies in the field of information systems thinking that could potentially be applied to the development of the generic framework. An evaluation of the hard and soft dichotomy that exists between approaches is discussed along with an evaluation of each methodology.

Chapter 3 introduced background research and principles that were to underpin the requirements for the generic framework. These were:

- Self-adaptive functionality;
- An understanding of autonomic computing and its inherent principles as discussed and reviewed by IBM [192]; and
- Control systems theory.

Chapter 4 provides a discussion of current and emerging thinking in the field of information systems through the form of a literature review. Recent years has seen a vast amount of research in this field. This chapter provides a discussion on the current position of information systems thinking and how these principles could be applied to situations where unpredictability and unexpected changes occur. The principles would need to be able to cope with that change, implement an appropriate resolution and do so without necessarily requiring assistance from a user.

Chapter 5 establishes the baseline requirements for the development of the generic framework. A discussion is included that looks at incorporating self-management, self-adaptation, and reconfiguration. Based on the literature review detailed requirements are then developed ready for the construction of the generic framework. In essence, the generic framework must contain functionality that can be utilised for:

- Detection of conflicts;
- Identification of conflicts;
- Classification of failure;
- Conflict resolution strategies; and
- Control rules.

Chapter 6 discusses the design of the generic framework and then establishes a baseline for its development in Chapter 7. The design of the generic framework incorporates the concepts of autonomy and self-adaptation and the requirement for these to play a significant role its functionality.



Chapter 7 provides a structured account of the development of the generic framework. Functionality of the methodological approaches chosen is evaluated and combined into the generic framework that is constructed based upon the established requirements.

Chapter 8 provides a detailed account of Merseyside Police, the organisation within which the generic framework would be tested. This chapter provides an overview of the organisation's position prior to the implementation of the generic framework. The structure of the organisation is discussed along with an introduction to the strategic aims of Merseyside Police. Chapter 8 discusses the implementation of the generic framework to the problem situation within Merseyside Police. Subsequently, a resolution is developed leading to the application of a new information system within the organisation. A discussion into the underlying issues in the problem situation is provided along with a detailed account as to how this is incorporated into the implementation of the generic framework.

Chapter 9 establishes a suitable testing environment for the implementation of the generic framework. This is conducted to ensure that accurate information can be gathered when assessing the successes and failures of the implementation of the generic framework.

Chapter 10 evaluates the information gathered from the defined testing environment and provides analysis of how the implementation went. Qualitative and quantitative information is assessed with the results demonstrated highlighting areas where the implementation of the generic framework has been successful and areas for potential improvement. A discussion is included which monitors the generic framework's adherence to its initial principles and how its functionality worked when called upon in Merseyside Police during its implementation.

Chapter 11 provides conclusions drawn from the study, along with a review of the contributions made and associated achievements. This chapter also looks towards the future and work that is being proposed through the continued use of the generic framework within Merseyside Police.

## **11.6 Discussion**

This thesis has focussed on the design and construction of a robust generic framework that can be implemented in complex problem situations with self-adaptive and autonomic concepts. The main contributions of this thesis incorporate the design and development of a generic framework able to adapt to changing environments.

Integration of self-adaptive and autonomic principles that allow for self-management and control are new contributions to the field of information systems. The control functionality of the generic framework contains the ability to control and reconfigure according to situations that arise. The ability to make appropriate repairs when necessary and update relevant policies and resolution strategies is incorporated into the generic framework design.

Functionality that allows the generic framework to detect conflicts and assess boundaries of norms within its environment is included in the design in accordance with autonomic principles. Linking the capabilities of two well respected methodological approaches to produce a generic framework that contains both hard and soft functionality is also a new contribution to the field.

Application of the generic framework in a real world problem situation is demonstrated in this thesis. The development of a new information system is explored with the results provided and evaluated.

## **11.7 Future Work**

It is imperative that the generic framework is implemented in new environments both in the public and private sector. To ensure the generic framework adheres to its requirements fully it must be exposed to different problem situations in order to see how it responds. The results of such implementations can then be reviewed leading to an evaluation of its performance and whether any further developmental work is required.



The key areas of future work are based around the further testing of the autonomic and self-adaptive capabilities of the generic framework through applying it to many varied environments. The purpose of the generic framework is that it can cope within any environment within which it is applied. Therefore testing the generic framework, and developing several iterations of that testing, is considered crucial to the future work.

In the context of Merseyside Police whilst the development of the generic framework, and subsequent implementation within the organisation, can be deemed a success through an evaluation there is outstanding work still to do.

The generic framework must be run through several further iterations in a testing environment to monitor how it is able to continuously adapt to changing requirements until the initial problem situation within which it was implemented is altered entirely.

Robust testing of the generic framework in Merseyside Police will continue in order to monitor how it copes in a rapidly changing environment. This will help prove its capacity to cope with change and backup information collected from its further iterations as outlined in the point above.

Merseyside Police has embraced the concepts of the generic framework and is keen to conduct further work to ensure that it keeps driving itself forward. In the case of Merseyside Police this includes continuing to develop the provision of information available within the organisation. As has been discussed in this thesis the introduction of GPRS enabled Blackberries is something that is being implemented. This will provide staff with dynamic accurate and relevant information whilst on patrol, thus removing the need to be inside a police station to gather required intelligence, which further develops the autonomic and self-adaptive functionality of the generic framework.

## **Glossary**

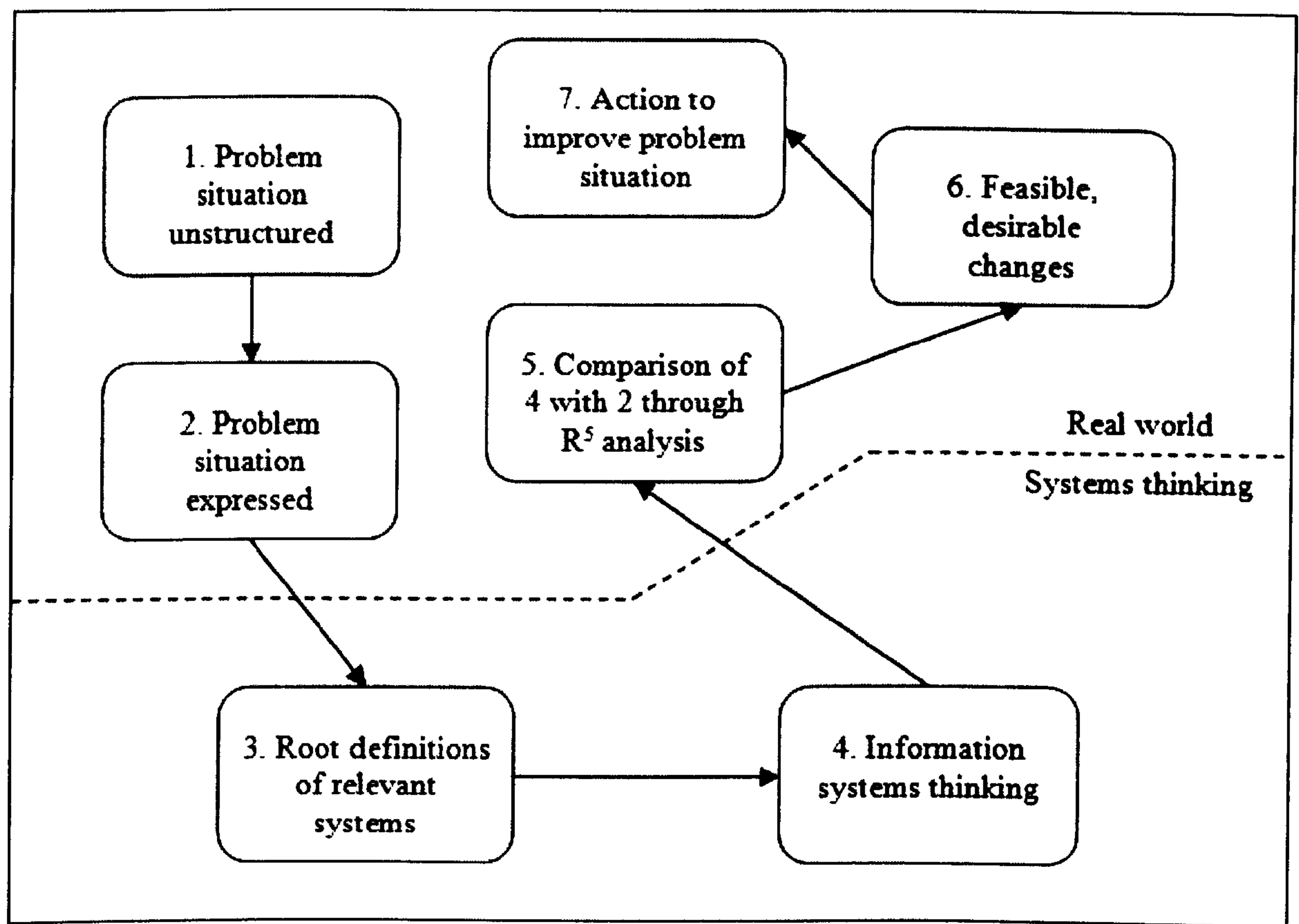
ADL	Architecture Description Language
ASB	Antisocial Behaviour
BCU	Basic Command Unit
BDI	Beliefs Desires Intentions
CATWOE	Customer, Actor, Transaction, Weltanschauung, Owner, Environment analysis functionality of Soft Systems Methodology
DARPA	Defence Advanced Research Projects Agency
EDA	Epistemic Deontic Axiologism
ETHICS	Effective Technical and Human Implementation of Computer-Based Systems
FCOU	Force Crime Operations Unit
GIS	Geographical Information Systems
GPRS	General Packet Radio Service
IRMA	Intelligent Resource-Bounded Machine Architecture
MPA	Merseyside Police Authority
NIM	National Intelligence Model
OGC	Office of Government Commerce
PCSO	Police Community Support Officer
SSADM	Structured Systems Analysis and Design Method
SSM	Soft Systems Methodology
VSM	Viable Systems Model



## **Appendix**

This appendix contains details user guidance for the generic framework and of information gathered and analysed in relation to the case study included as part of the research presented in this thesis.

**Generic Framework User Guide**

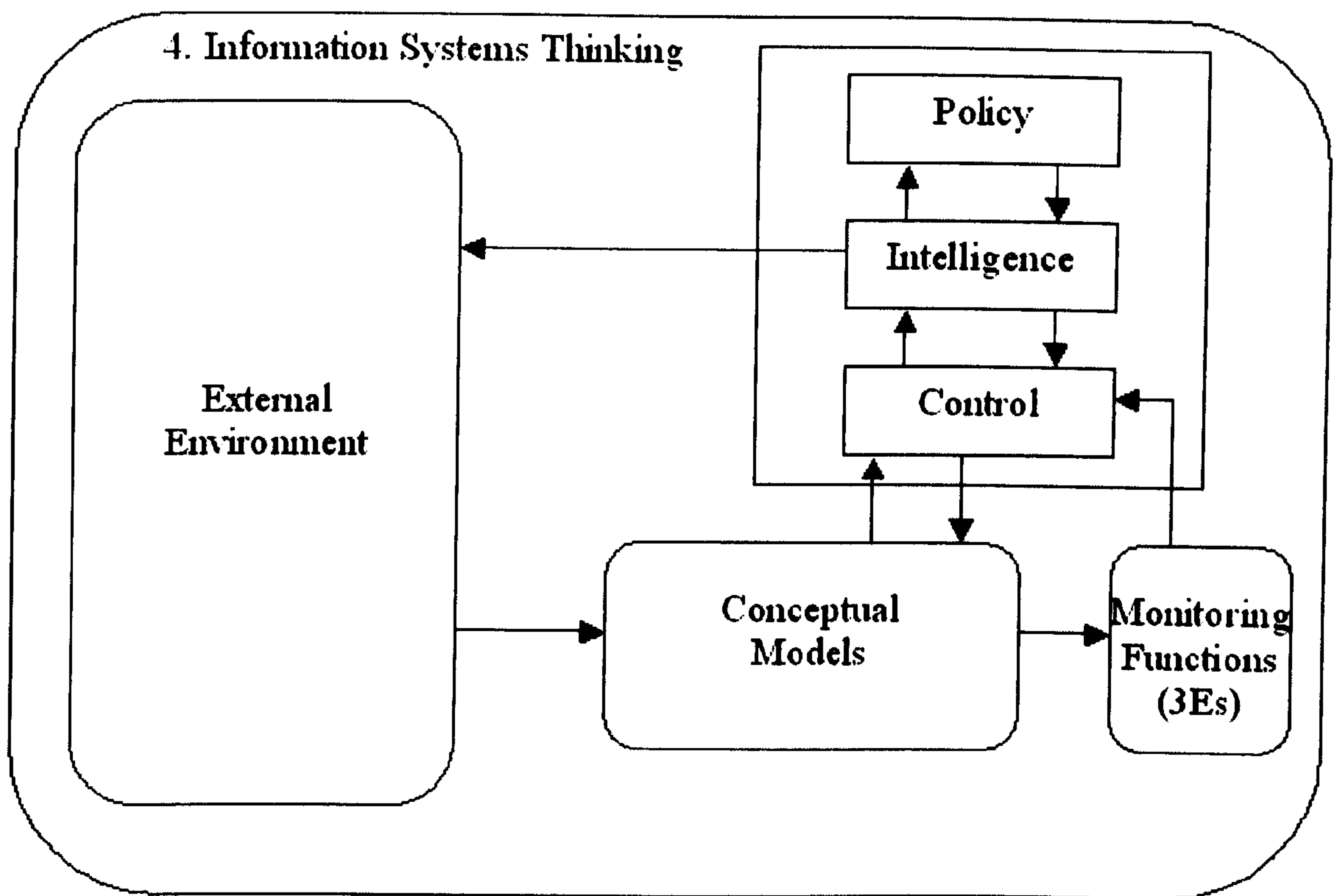


The generic framework is designed to be simple to use and can be applied by anyone to any given problem. The following seven steps will lead the user towards implementing a new approach to resolve problems being experienced.

The Seven Steps of the Generic Framework

- Step 1 – The Problem Situation.* This is where the analyst derives knowledge of the organisation, how it works and what the working procedures are at present.
- Step 2 – The Problem Situation Expressed.* This is where the analyst writes down the knowledge gained from the previous step. This must be written down in a way that people in the problem domain can understand.
- Step 3 – Root Definitions of Relevant Systems.* This step is where the analyst tries to define/create a system at a conceptual level that will address the problem in the problem domain.
- Step 4 –Information Systems Thinking.* Step 4 is where a check is done to validate the root definitions. The basic ‘whats’ of the process come from the root definitions and describe what the process should be like.





*Step 5 – Comparison of Step 4 with Step 2.* Here the analyst compares the conceptual models (Step 4) with the real world (Step 2) to see where they differ and are similar. This is done through following the five steps of the R<sup>5</sup> model as follows:

- Resources;
- Rules;
- Responsibilities;
- Regulations; and
- Recommendations

*Step 6 – Feasible, Desirable Changes.* This step is where the analyst identifies feasible and desirable changes e.g. introduction of a new information system.

*Step 7 – Action to Improve the Problem Situation.* There is where recommendations for taking actions to improve the problem situation are made.

## How to Use the Seven Stages of the Generic Framework

*Step 1* – The unstructured problem situation is the initial information gathering phase of the generic framework. Do not make any assumptions about what the problem is and concentrate on writing down all pertinent facts to describe the whole situation as it currently is.

*Step 2* – The Problem Situation expressed. This is a structured definition of the problem situation and should follow the steps below:

- Use a wide range of people with roles in the situation
- Interest groups
- Collect as many perceptions of the problem as possible
- Features may include:
  - Structure... physical layout, power hierarchy, reporting structure
  - Communications... informal & formal
  - Process... operations, monitoring, decision making & control
  - Climate... relation between structure and process
- Performance
- Reflect on world views
- Identify conflicts

Rich Pictures as part of Step 2.<sup>3</sup>

An approach by which subjective interpretation and understanding of messy situations is achievable and consequently transferable for other people's subjective consumption, is the rich picture approach of Checkland. A rich picture is one that expresses, in as rich a manner as possible, a person's general appreciation of a perceived problematic situation. The idea developed as an integral part of Checkland's soft systems methodology, although the approach has been applied to a wide range of situations outside the Human Activity Systems of SSM.

---

<sup>3</sup>The expression of a problem situation, compiled by an investigator, often by examining elements of *structure*, elements of *process* and the situation *climate*.



Essentially, the development of a rich picture parallels brainstorming, but represents the ideas in pictures rather than words. It is like a large cartoon representation of a situation in non-system terms.

The upshot is that the rich picture is completely flexible allowing the analyst freedom to develop a graphic depiction of the current situation. This, in turn, provides a structured understanding of the current situation and leads the analyst towards understanding the problems being experienced.

*Step 3 – Root Definition of relevant systems defines the situations being experienced and what appears to be at the root of the problem.*

A Root Definition is a concise, tightly constructed description of a Human Activity System which states what the system is; what it does is then elaborated in a conceptual model which is built on the basis of the definition. Every element in the definition must be reflected in the model derived from it. A well formulated root definition will make explicit each of the elements in the mnemonic **CATWOE**.

C (Customers)... who are the direct victims or beneficiaries of the transformation?

A (Actors)... who would do these activities?

T (Transformation process)... what input is transformed to what output?

W (World)... selected world view; a system for?

O (Owner)... who could abolish this system?

E (Environmental constraints)... what does this system take as given?

A general Root Definition embodying CATWOE might take the following form:

*“A (..O..) owned system which, under the following environmental constraints (..E..), transforms this input (..input..) into this output (..output..) by means of the following major activities (...), the transformation being carried out by these actors (..A..) and directly affecting the following beneficiaries and/or victims (..C..). The world view which makes this transformation meaningful contains at least the following elements (..W..).”*

No Root Definition can ever provide a unique description of any actual manifestation of a Human Activity System. It will always be only one possibility out of a large number. A Root Definition will only be a meaningful description of the relevant system according to a particular world view.

**Practical Guidelines for Developing a Root Definition:**

- Describe an interpretation of what exists, e.g. 'A brewery owned system for...' 'not' A pub is a building...'
- Focus on a Human Activity System<sup>4</sup> not a physical system.
- Customers are directly affected by the transformation not several stages removed.
- Avoid complex Root Definitions, adopt only one transformation per Root Definition and one transformation to reflect one world view.
- Use CATWOE

#### *Step 4 – Making and Testing Conceptual Models.*

The following steps should be considered when looking to develop a Conceptual Model for the given problem situation and system being reviewed.

- A Conceptual Model is an account of what the system does in order to be what it is
- Conceptual Models are based on activities with elements as verbs
- In particular, Conceptual Models are based on the minimum and necessary set of activities as defined in the Root Definition...and nothing else.
- Activities should be structured into a logical sequence.
- The first basic Conceptual Model could consist of a main activities to give a low resolution model, which should only be resolved into further detail as and when necessary.
- For each activity or decision ask the questions:
  - What data are needed?

---

<sup>4</sup>Human Activity Systems describe some purposeful human activity.



- What is the data content?
- What are the sources of data?
- How frequently is the data needed?

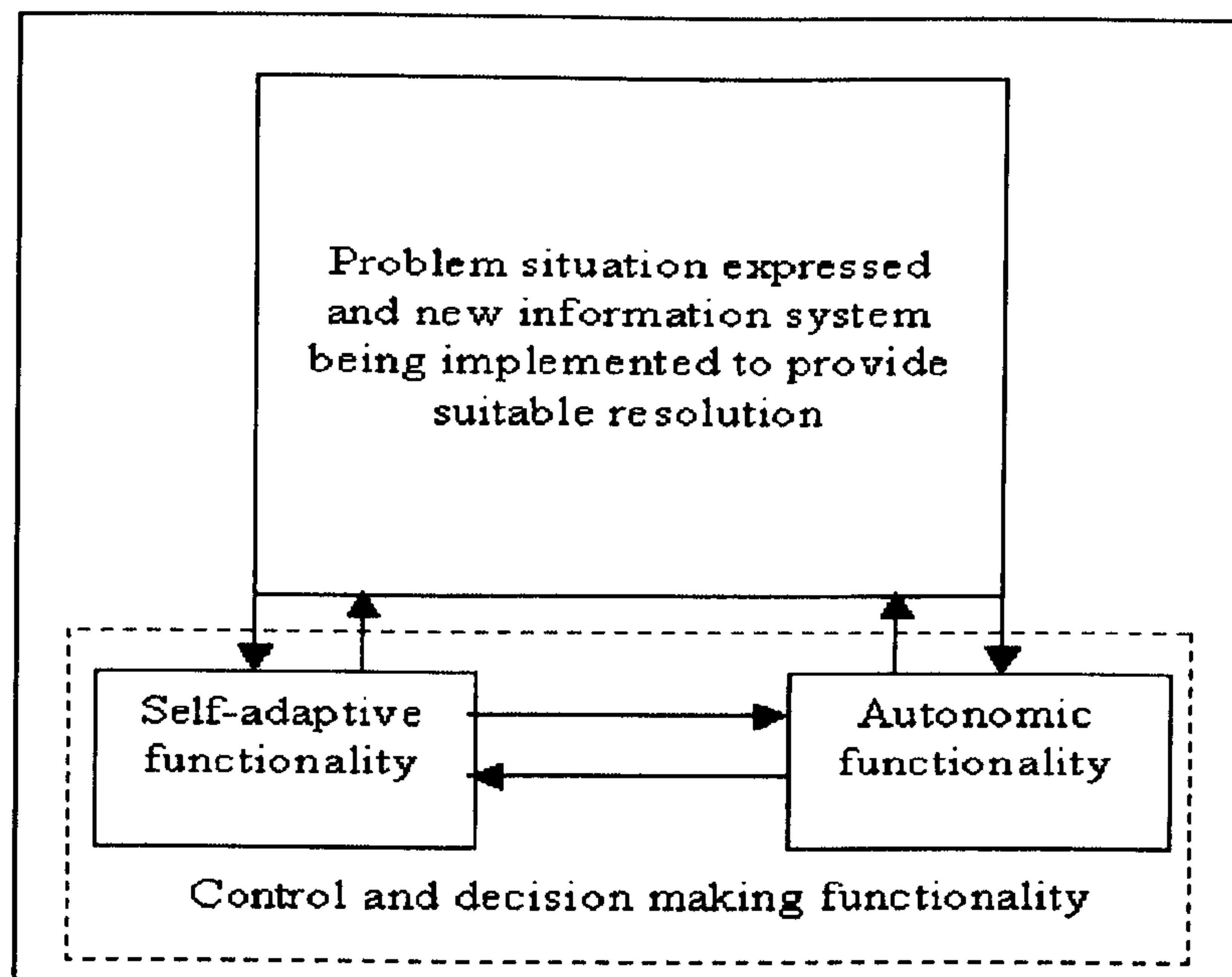
The following Laws should be considered in developing Conceptual Models:

***The Law of Conceptualisation:*** *a system which serves another cannot be defined and modelled until a definition and a model of the system served are available.*

***The Law of Model Building:*** *models of Human Activity System must consist of structured sets of verbs specifying activities which actors could directly carry out.*

Practical guidelines for developing Conceptual Models are detailed below to assist the analyst to construct something accurate and robust:

- Developing Conceptual Models involves a process of iteration between the Root Definition and Conceptual Model, refining each until a reasonable model is reached.
- Avoid too sparse (simple) a Root Definition by using CATWOE
- Language – Express activities as verbs. The presence of a noun may indicate that a real-world constraint (how the activity is done) is being imposed on the model.
- Method – Develop the ‘minimum, necessary and sufficient activities’ for the Conceptual Model by identifying its output(s) and in reverse order, the activities required to produce these output(s).



The high level Conceptual Model should follow the form shown above. It represents the activities taking place with control and decision making capacities being established to ensure that appropriate governance and corrective measures can be enacted.

When the Conceptual Model has been developed it should have taken into consideration the following:

- The model could be a compilation of ‘management’ components required by a system that is capable of purposeful activity.
- This Human Activity System could therefore include:
  - An on-going purpose or mission.
  - It may relate to goals, objectives and ideals.
  - A measure of performance
  - Decision-making process
  - An environment
  - System Boundaries
  - Resources
  - It has stability and viability



These points allow an analyst to reveal inadequacies in either the Conceptual Model or Root Definition.

*Step 5 – Comparison of Conceptual Model within the real world situation.*

The difference between the Conceptual Model and the real world should be sufficient to generate debate and further enquiry, but not too great to seem unrealistic to the point of terminating further enquiry.

Too small a difference will seem unreasonable for the effort put into the study and too large a difference would probably seem unrealistic and ‘idealistic’.

There are four ways of comparing the Conceptual Model with the real world. These are as follows:

- Use the Conceptual Model to generate questions.
- Use the Conceptual Model to reconstruct a sequence of events in the past and compare with what would have happened if the relevant Conceptual Model had been implemented.
- In some cases, major strategic questions will be raised about present activities; Why do this at all? One can concentrate on the features of the Conceptual Model that are especially different from the present and ask why?
- Use the same form of model in the Conceptual Model and in the model of ‘what exists’ and by overlaying models, reveal mismatches. One can also ask ‘What Root Definition is implied by the existing system’ where a significant difference exists.

*Step 6 – Systemically desirable and culturally feasible changes.*

The definitions to be taken into consideration for Step 6 are as follows:

- Systemically desirable – if the change results from a ‘well conducted’ application of the SSM.

- Culturally feasible – if those involved in the situation can express ways in which the change can be made.

The analyst must also consider three types of change:

- Structural – static elements such as organisation and reporting structure.
- Procedural – dynamic elements.
- Attitudinal – expectations and attitudes.

*Step 7 – Action to improve the problem situation.*

Having determined the systemic desirability and cultural feasibility the analyst can then work towards implementing the appropriate resolution to the problem situation.

It should be noted that the generic framework is flexible and allows the analyst to make several iterations, back-track across various Steps as and when required, and also miss some Steps completely should the need arise.

The analyst is requested to follow the principles put forward in this guide to assist them with determining appropriate resolutions and also ensure that the information system they put in place is capable of working in a self-adaptive and autonomous way.



## **Business Benefits of Niche**

Since the implementation of Niche RMS for crime recording commenced on 4<sup>th</sup> June 2007 various performance measures have been put in place to monitor its success or otherwise since its introduction to the Force.

The measures included in the business benefits analysis are as follows:

- Crime and detections
- Length of time to detect crime
- Custody airlocks
- CCRB waiting times
- CCRB call handling times
- Named suspects
- Reclassifications reviewed
- Reclassifications approved
- Time to taken to record crime
- 28 day crime
- System performance – planned and unplanned downtime
- Data quality
- Occurrence closures

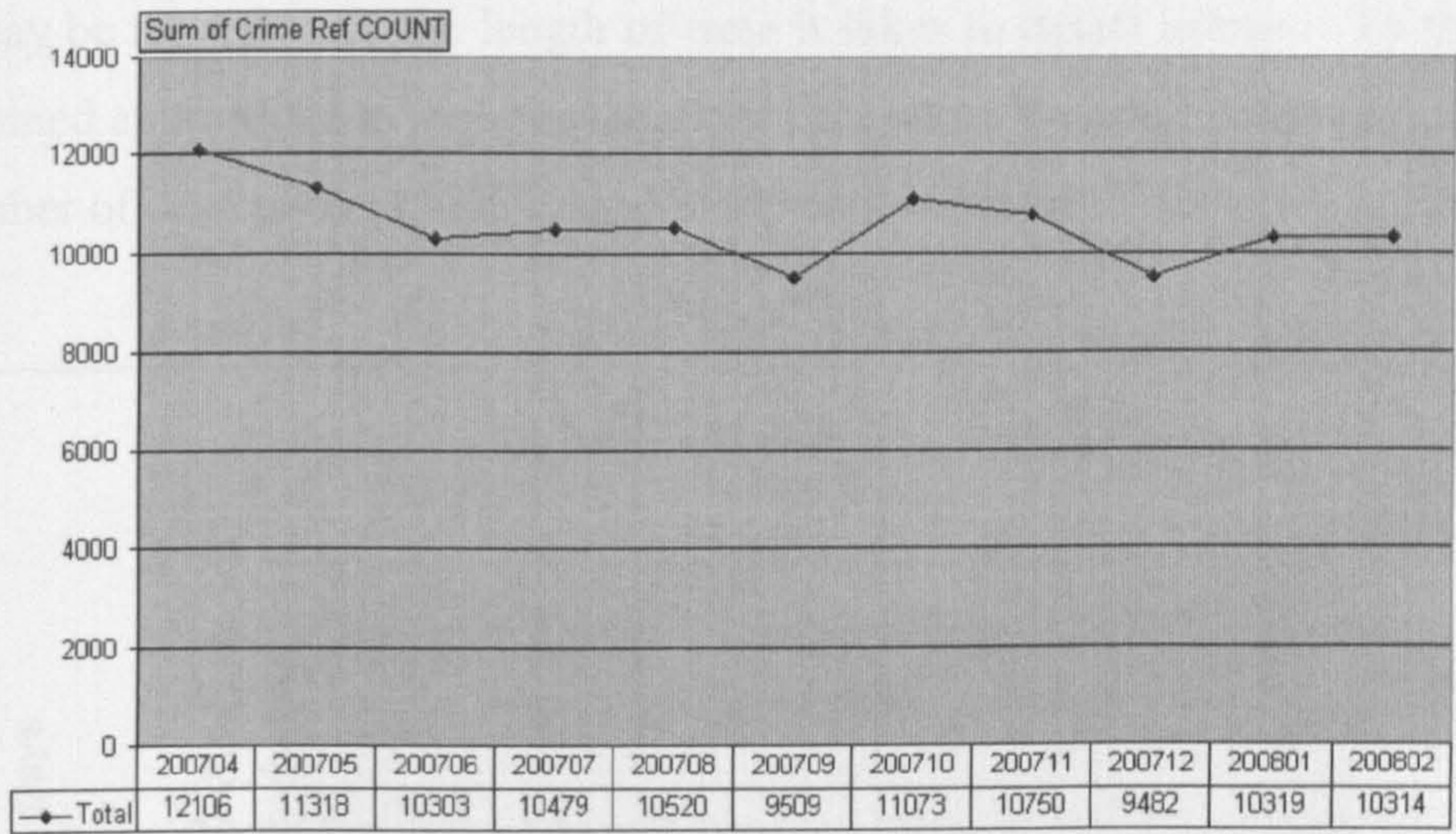
The information demonstrated through the above measures provides details of the impact Niche RMS has had on the Force since its implementation for crime recording.

### **Crime and Detections**

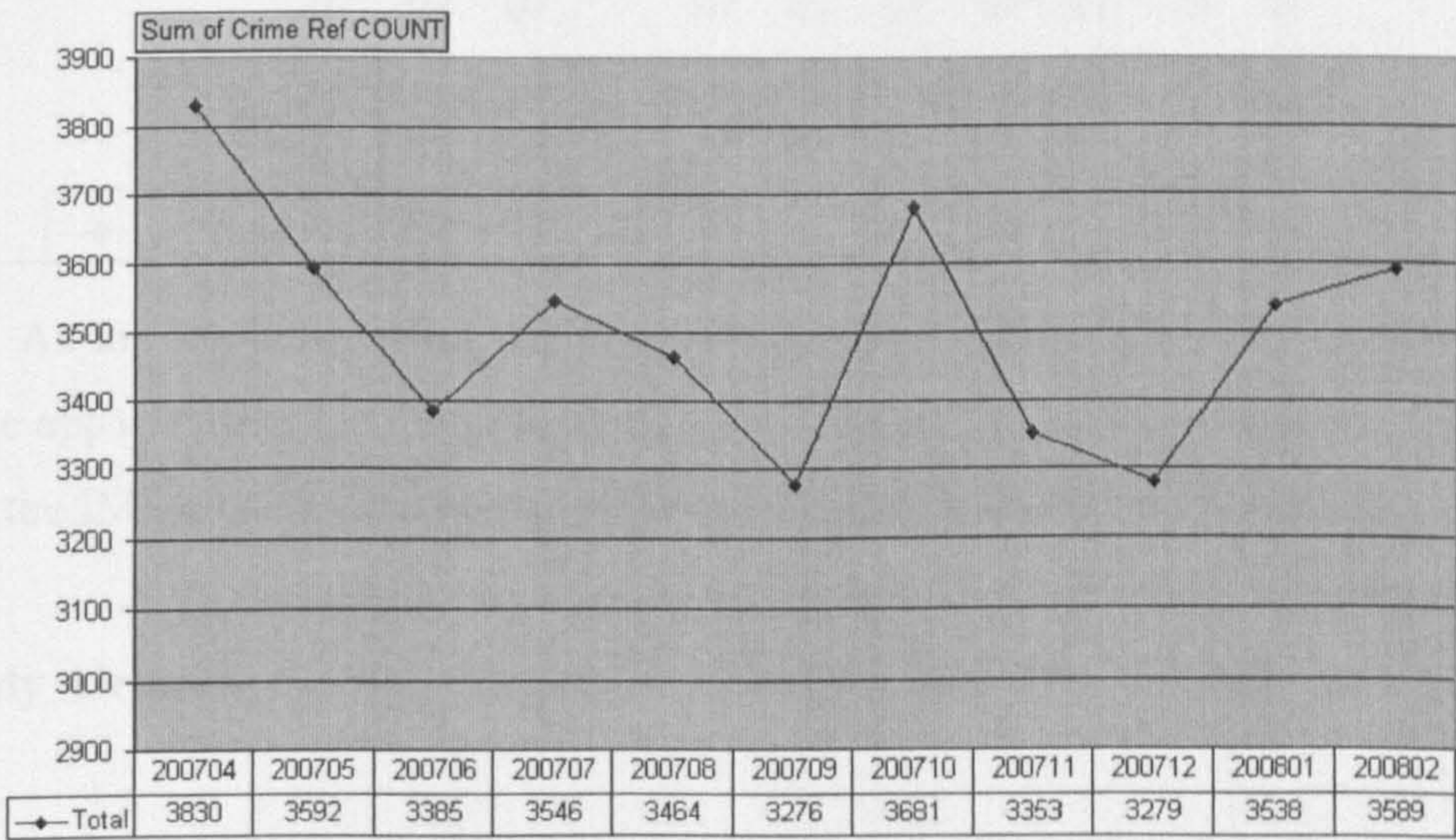
The most commonly known crime and detection performance indicators can be located within the Daily Crime. This product, developed and published by Information Development, is posted on Galleries just after 07:00 each morning.



The crime and detection information collected covers Force performance for each BCU. This is deemed to be the most appropriate indicator for monitoring crime and detection related performance.



The chart above demonstrates BCU performance trends that are consistent with previous years. That said, the actual number of crimes is lower than is previous years.

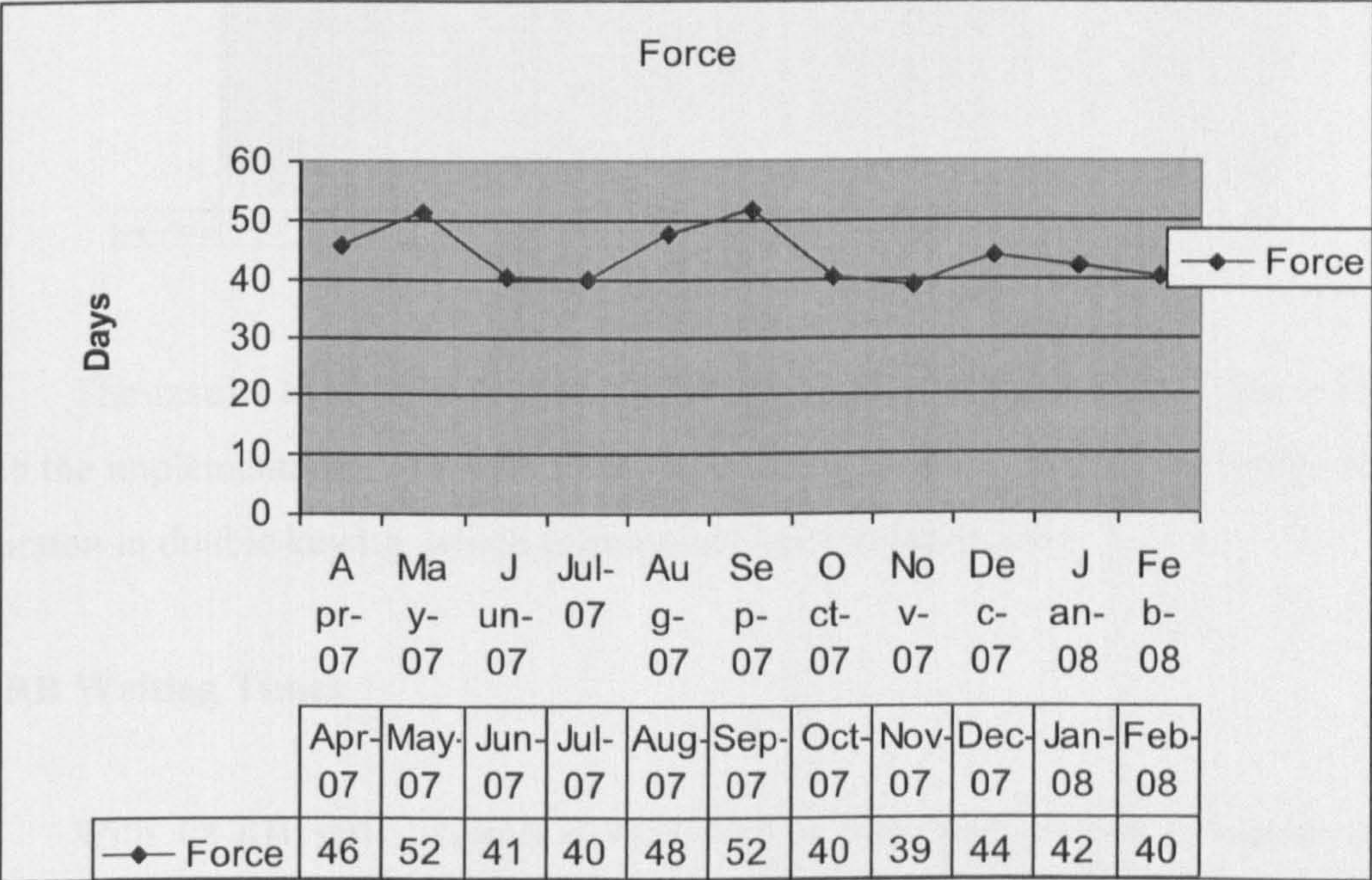


Performance with regards to detections is also consistent with previous years but is showing a rise in the number of actual detections recorded. The fact that there was no negative impact to Force performance, and that the number of detections had risen on previous years, reflects positively on the implementation of Niche RMS for crime recording.



Length of Time to Detect Crimes

With staff familiarising themselves with Niche RMS it was anticipated that there may be an effect on the length of time it takes to detect crimes. To this end it was deemed appropriate to monitor length of time taken to detect crimes in addition to the number of detections as highlighted in the Daily Crime.



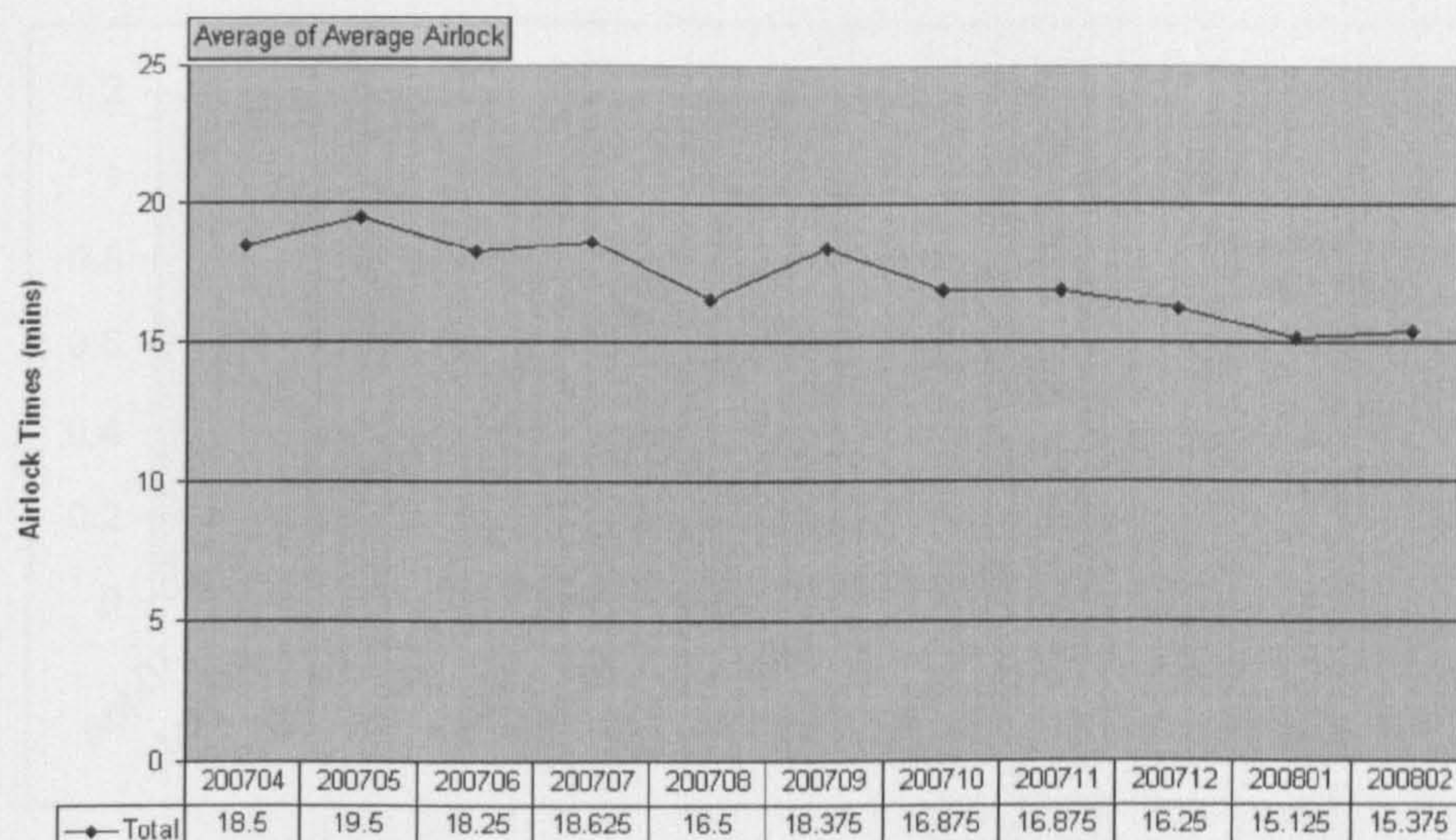
As the implementation of Niche RMS reached phase 3 in November 2007 it became apparent that the main impact on the length of time taken to detect crime was due to the IMS closure plan being instigated rather than Niche RMS itself.

Custody Airlocks

To establish what effect the introduction of crime recording on Niche RMS has had across the Force it is proposed that custody airlock times are assessed. Although Niche RMS will incorporate crime recording it is possible that this could have a knock-on effect with performance in Force custody suites.

The most appropriate way of monitoring performance within custody suites is through airlock times as this gives an indication as to how long it takes to process prisoners.





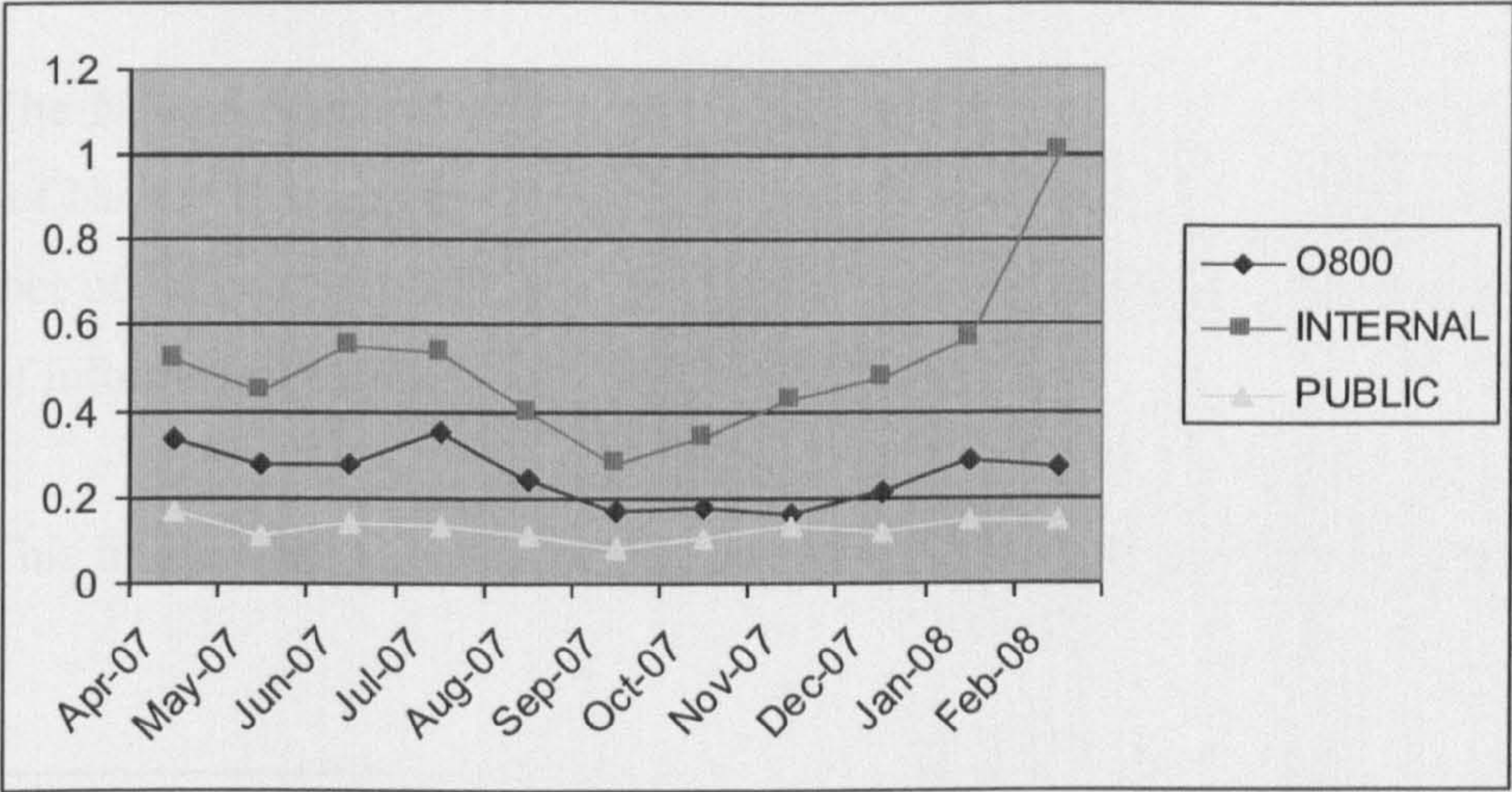
The results show that in most BCUs the average airlock times have reduced since the implementation of Niche RMS for crime recording. In part this is due to the reduction in double keying, which is discussed later in this paper.

### CCRB Waiting Times

With CCRB staff playing a vital role in recording crimes it was deemed essential that their performance be monitored. The CCRB routinely gathers information on average call duration and waiting times and it was proposed that this be used to monitor performance in relation to crime recording. The implementation of Niche RMS for crime recording was to be monitored through gathering management information in relation to average call durations and waiting times.

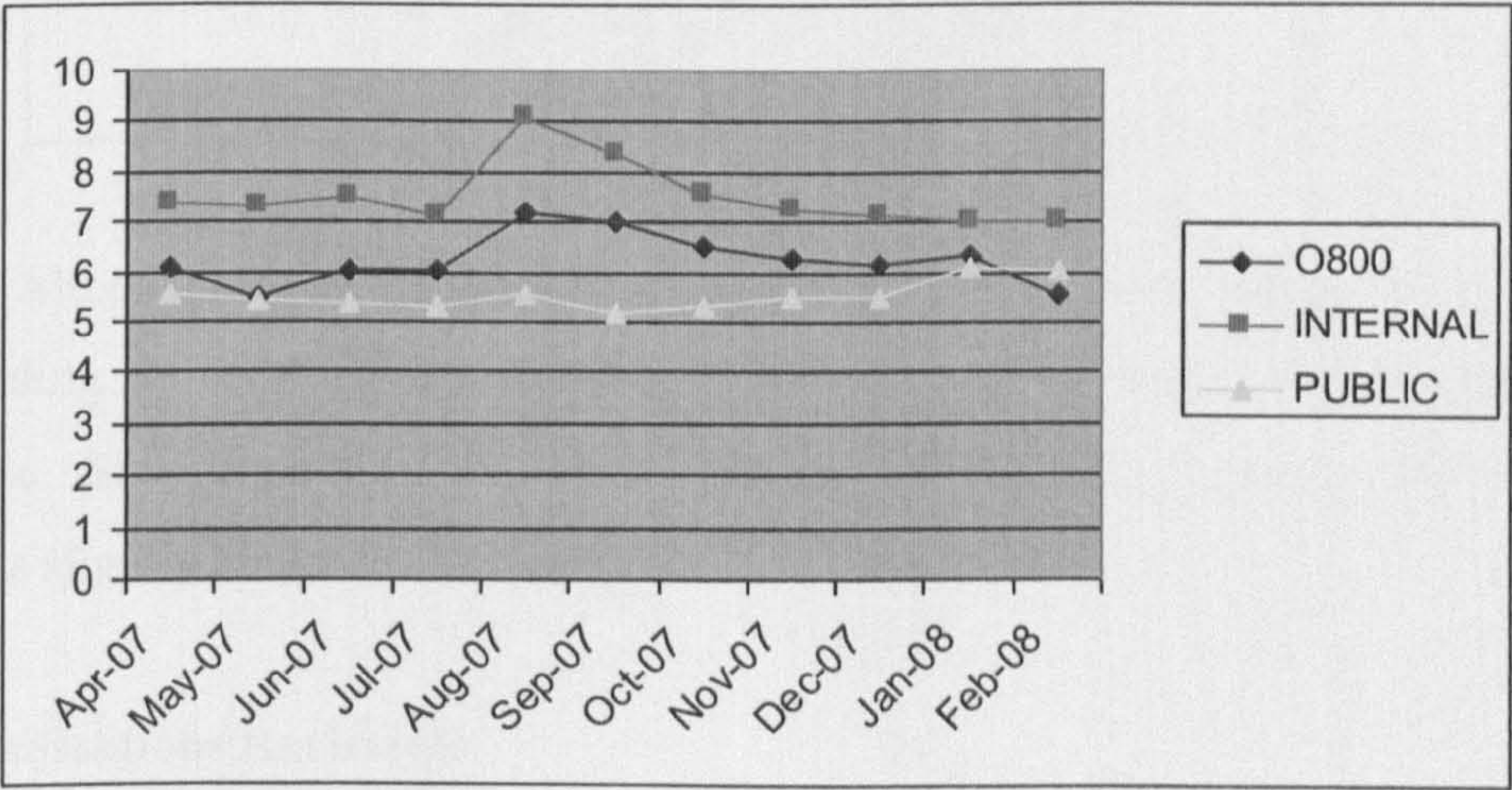
The main performance indicators for CCRB call handling cover 0800, internal and public calls. The intention is for the CCRB to answer 90% of public calls within 30 seconds, 80% of 0800 calls within 30 seconds and 80% of internal calls within 60 seconds.





Despite the rise in average call waiting times for internal calls, which Chief Superintendent Lewis is looking into, the overall averages for 0800 and public calls is less than before Niche’s implementation.

**CCRB Call Handling Times**



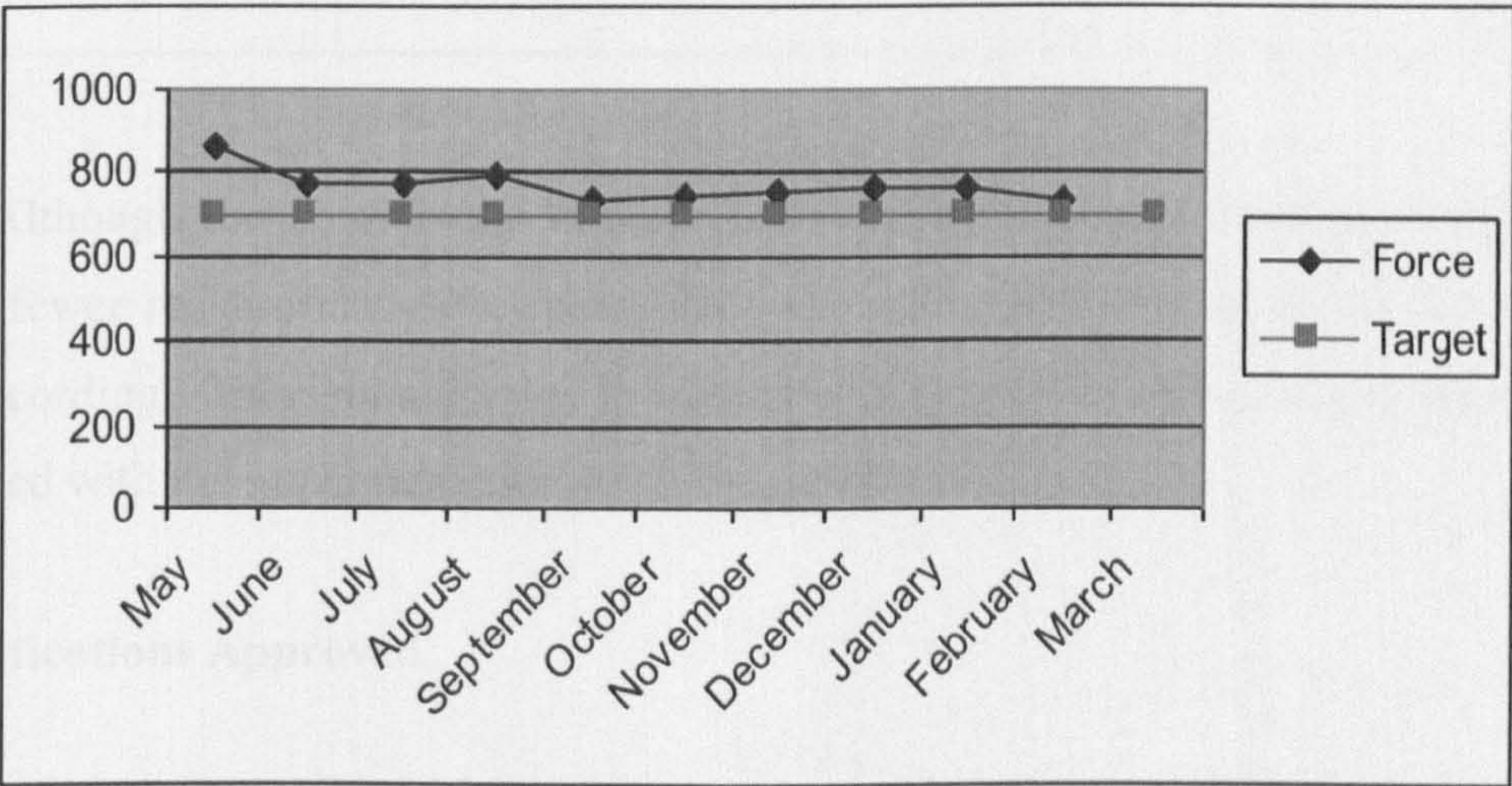
As with average call waiting times it can be seen that average call handling times in the CCRB are improving. It can be seen in the above chart that the average call handling times have increased slightly for 0800 and internal calls. Calls received from the public have actually reduced slightly since the implementation of Niche RMS for crime recording.



**Named Suspects**

The Named Suspects Policy states that the Force should aim to reduce the number of Named Suspects by 20%. In line with this reduction it was proposed that the number of Named Suspects be used as a performance indicator for monitoring the quality of information that is recorded on Niche RMS.

This information is collected on a monthly basis from the BCU Commanders Toolkit.

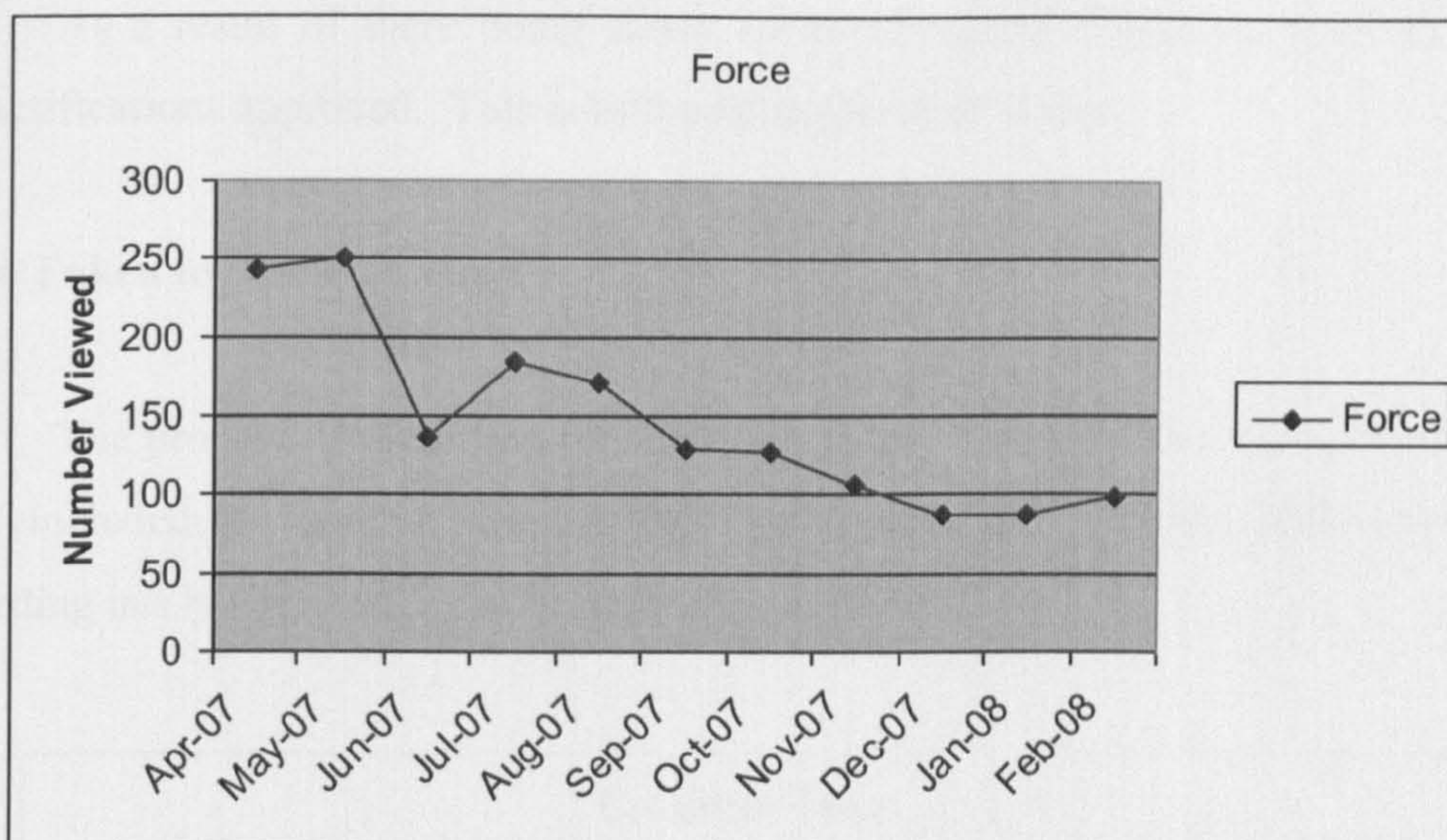


Although it can be seen the Force is above its overall target the number of outstanding Named Suspects is closer to that target than prior to the implementation of Niche. Since June 2007 the number of outstanding Named Suspects has fallen and now sits slightly above the Force target.

**Reclassifications Reviewed**

The reclassification rates are currently collated by the CCRB Audit Team on a monthly basis. The reclassification process on Niche RMS is performed manually and monitored by the CCRB, and as such, will report on the reclassification that have taken place in the previous month by BCU.

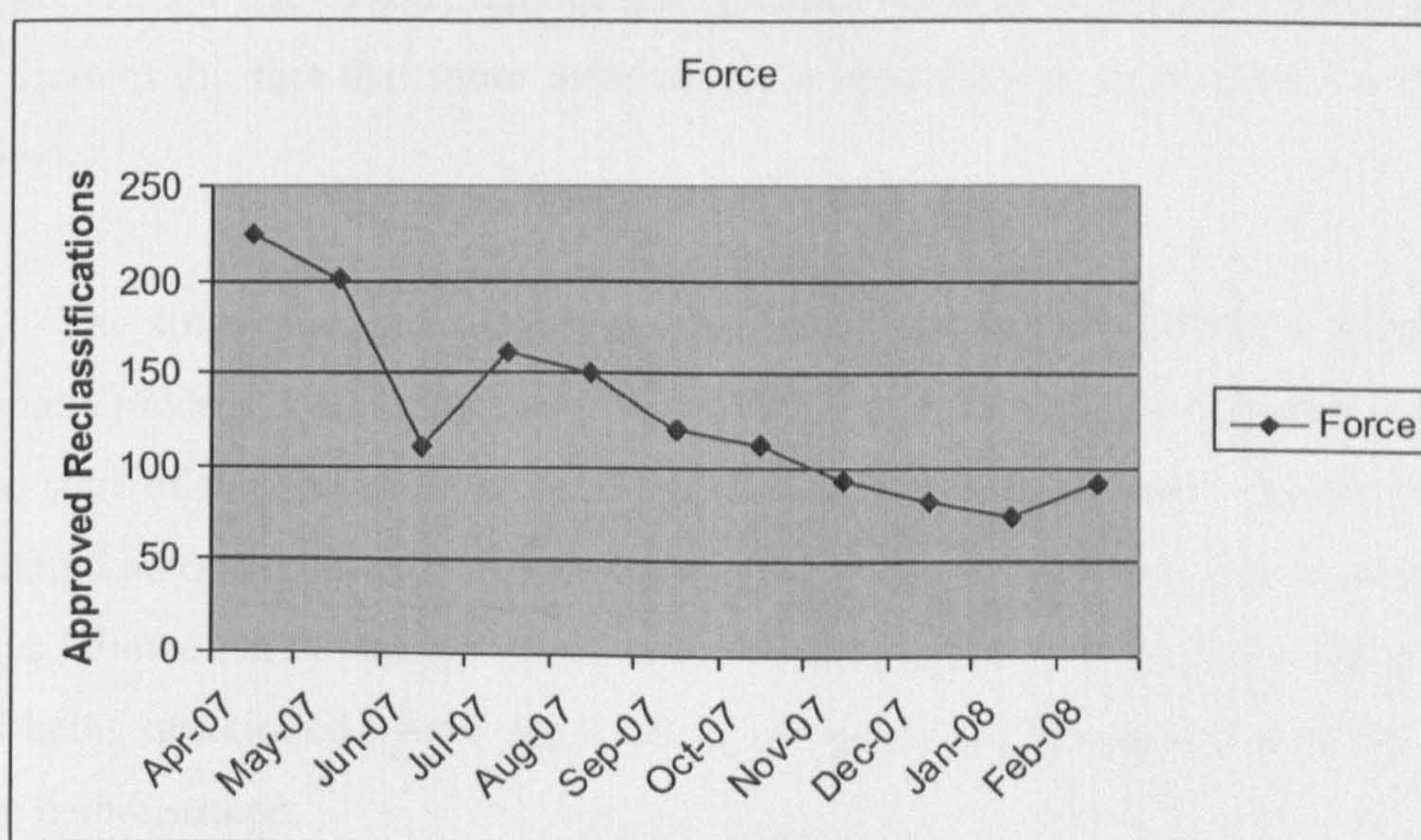




Although the chart looks rather cluttered it can be seen that each BCU is making fewer reclassification requests since the implementation of Niche RMS for crime recording. This is mainly due to improved data quality and recording practices introduced with the implementation of Niche RMS.

### Reclassifications Approved

Following on from the previous measure the number of reclassifications approved by the CCRB is also monitored.

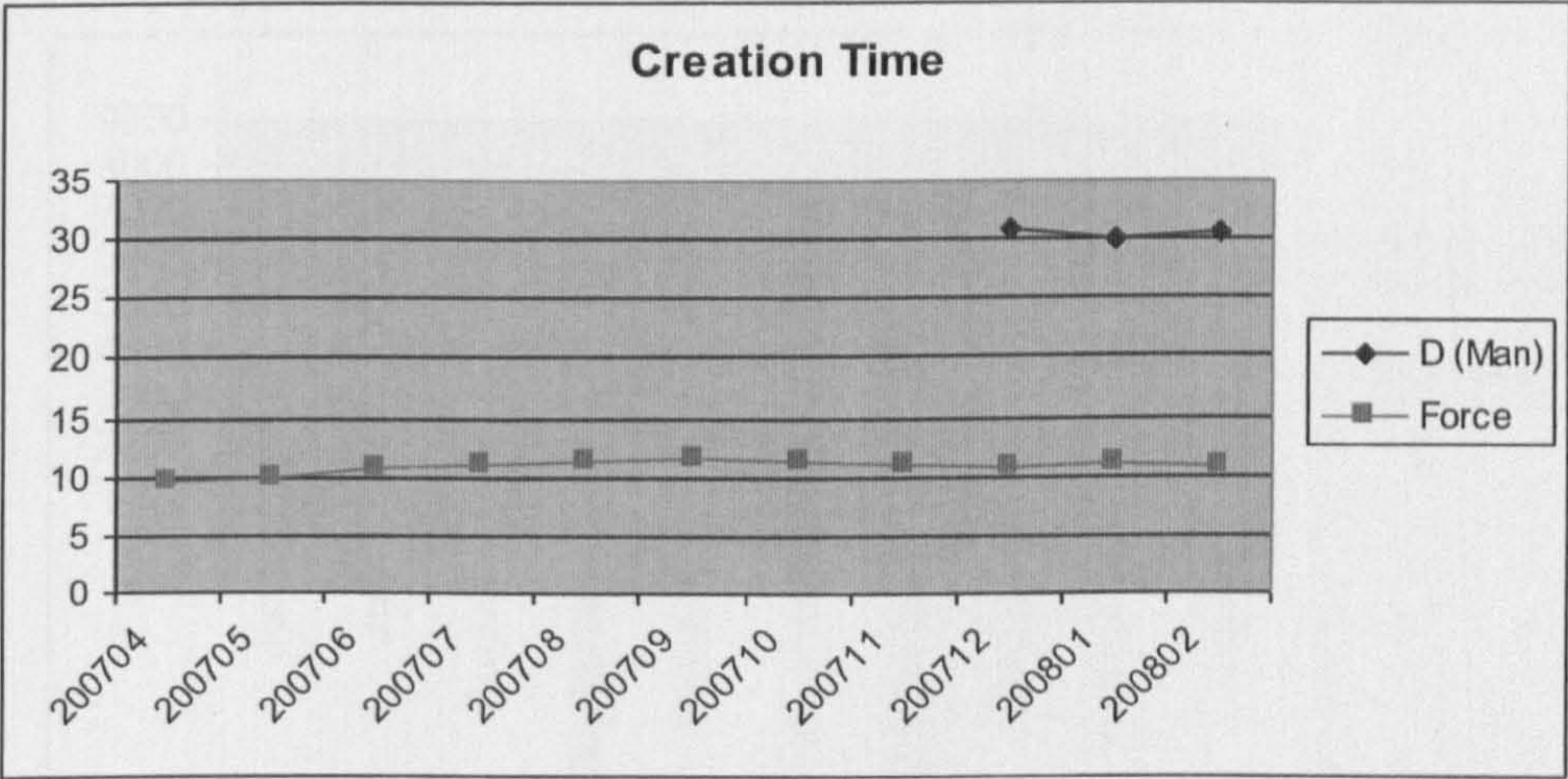




As a result of there being fewer reclassifications requested there are fewer reclassifications approved. This is reflected in the chart above.

**Time Taken to Record Crime**

The process of recording the amount of time taken to create a new crime has been included to monitor whether the implementation of Niche RMS for crime recording has had any effect on performance in this respect.



The chart above demonstrates a slight rise in the average time taken to record a crime. This is due to staff familiarising themselves with Niche RMS and also takes into account the fact that more information is recorded for each crime on the new system.

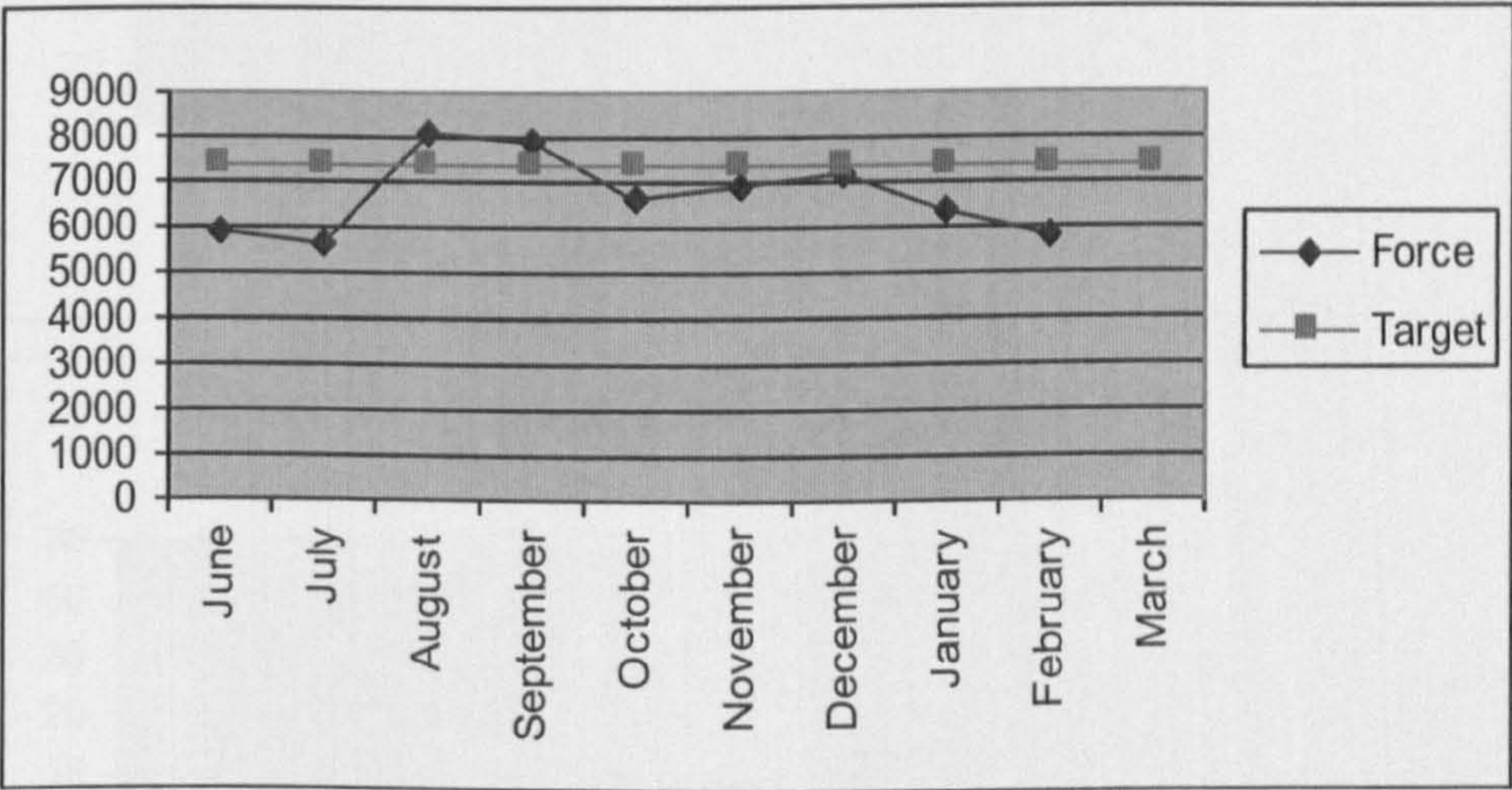
The study into self recording times in St Helens led to both a manual and automatic process being conducted. The CCRB staff create a new occurrence number at the start of the recording process and end it by sending a task. This process is monitored automatically through date and time stamps. However, this process is not always followed in St Helens, which led to the average time for record creation in that BCU being questioned. As a result a manual process was instigated with the results above demonstrated.



**28 Day Crime**

With staff familiarising themselves with Niche RMS Crime it was anticipated that there may be an effect on the length of time it takes to investigate a crime. It was therefore appropriate to measure the number of crimes that are over 28 days old.

Currently each BCU has a key performance indicator as to how many crimes over 28 days they should have. Information Systems currently collate this information on a daily basis and this is published via the BCU Commanders Toolkit.



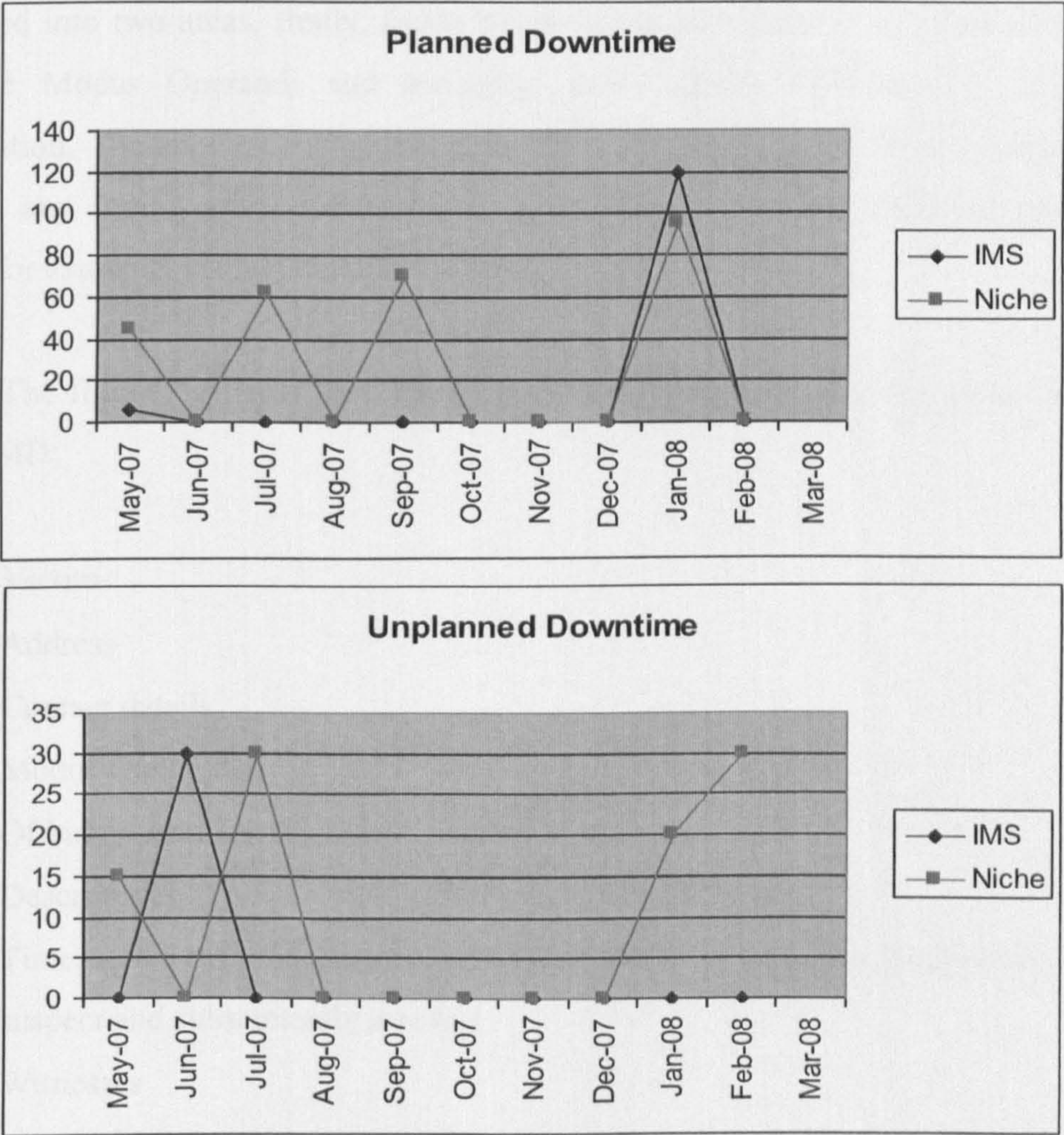
In much the same way that a fall in the number of outstanding Named Suspects was observed 28 day crimes has experienced a reduction too. After Liverpool North went live with Niche RMS in August 2007 an increase in 28 day crimes was highlighted as IMS crimes were not being filed in time due to staff familiarising themselves with the new system. The subsequent instigation of the IMS closure plan saw a drop in 28 day crimes.

**System Performance – Planned and Unplanned Downtime**

When attempting to measure the improvement in our service provision, it is essential that the reliability of the application be measured.



Currently, Information Systems can supply data in relation to planned and unplanned downtime on IMS. Following the implementation of Niche RMS for crime recording, Information Systems will record the reliability of the system.



Since the implementation of Niche RMS for crime recording the system has undergone some routine planned maintenance and upgrade work. As with any new system this is considered normal practice and has not caused any concern for the IS department. The unplanned downtime was also consistent with what would normally be expected for the introduction of a new system. That said, in January and February 2008 two periods of unplanned downtime were experienced. These periods of downtime were due to server related issues and not down to Niche RMS itself.



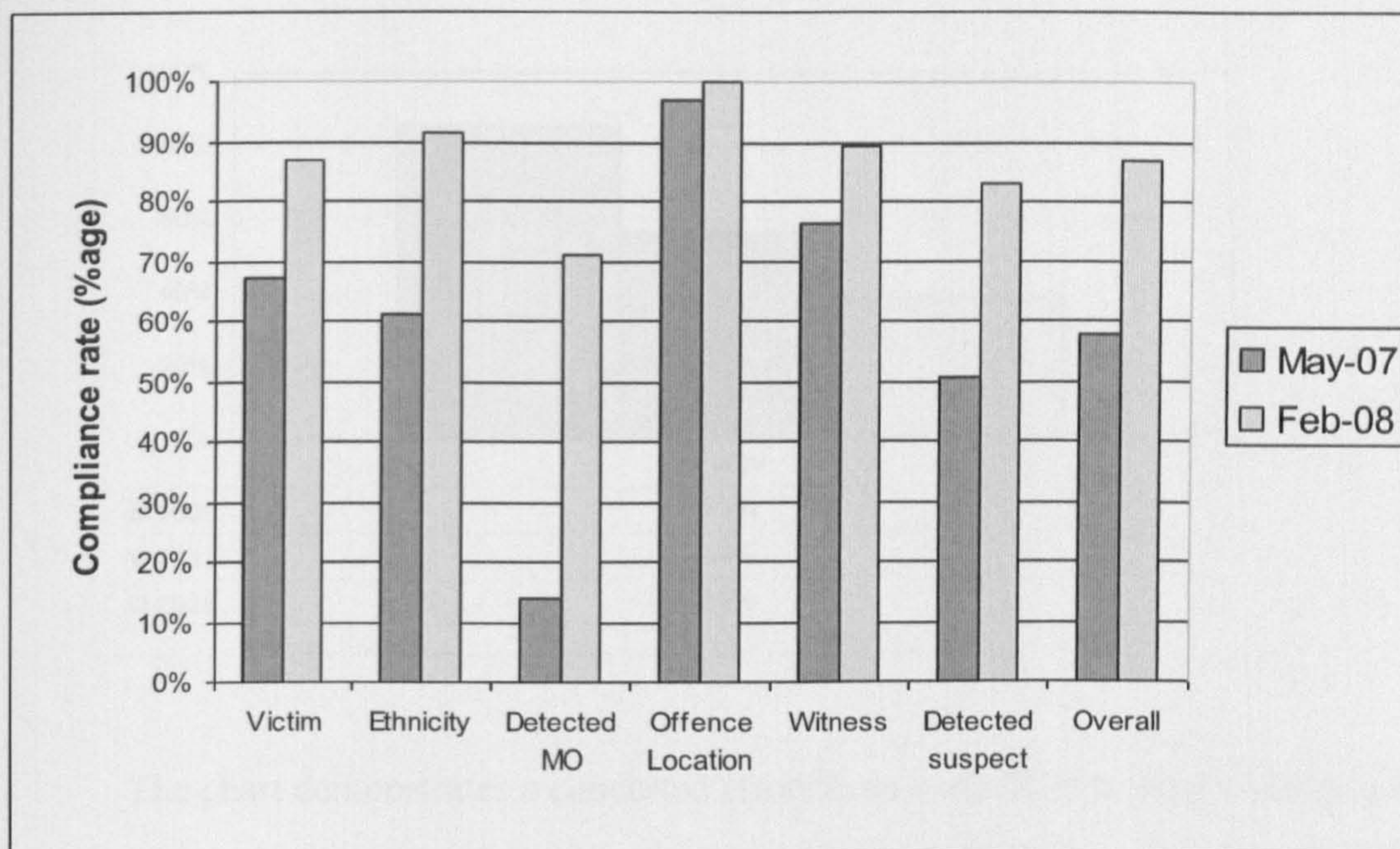
## Data Quality

Following implementation of Niche RMS, in order to identify an improvement in compliance there are a number of fields that will be measured. These will be separated into two areas, firstly, fields which will assist in securing a detection, for example Modus Operandi and secondly, fields which will support case file compilation. If compliance is measured on the latter, then we will be in a position to identify any improvement within the CJX exchange following implementation of crime, for example, contact details of witness.

The following fields have been identified and submitted to the Data Quality team, IMD:

- Victim
- Address
- Contact details
- Modus Operandi
- Offence Location
- Descriptives
- Time, as the difference in the time taken for an individual to be classified as a suspect and subsequently arrested.
- Witnesses
- Address
- Contact Details
- Current Officers
- Officers Location
- Pre Charge Case File Assembly (including typed statements) - Wirral only.



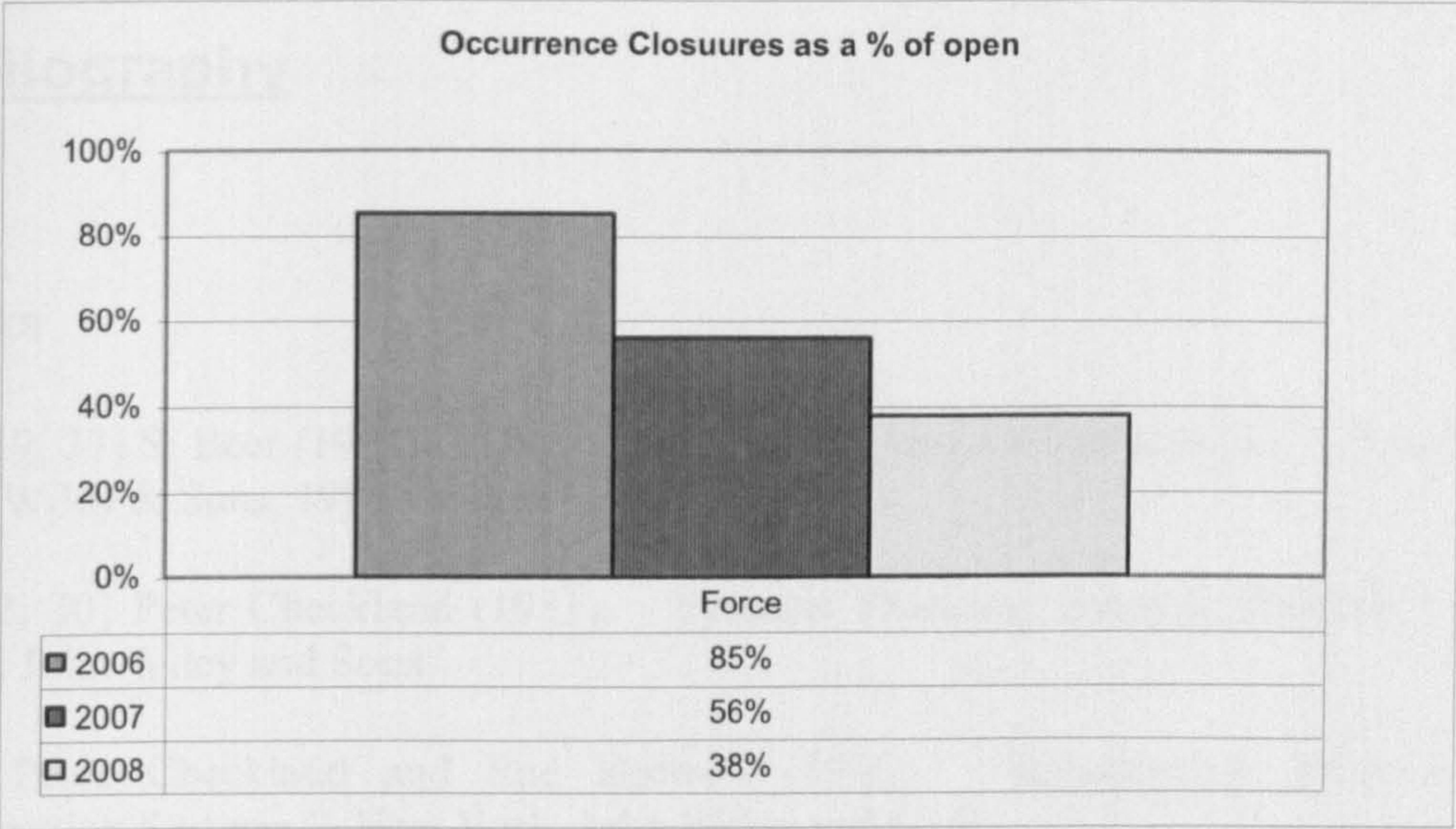


The results of the data quality investigations over a period of several months has shown an increase in the information recorded across all BCUs. Whilst it is acknowledged that some areas of improvement are required each of the performance indicators provided by the Data Quality team at Binns Road highlight an improvement when compared to IMS.

### Occurrence Closures

Following on from the investigation into data quality it was highlighted that occurrences on Niche needed to be closed once they had been finished with to ensure MoPI compliance. The following chart highlights BCU performance in this regard.





The chart demonstrates a concerted effort from each BCU to start closing open occurrences. At the time of writing this paper most outstanding occurrences from 2006 and 2007 had been closed.



# **Bibliography**

## **Books**

[18, 19, 33] S. Beer (1994). “Diagnosing the System for Organisations.” New York, John Wiley & Sons, 1994 version.

[3, 12, 30] Peter Checkland (1981). “Systems Thinking, Systems Practice.” New York, John Wiley and Sons.

[15] Peter Checkland and Sue Holwell (1998). “Information, Systems and Information Systems.” New York, John Wiley and Sons.

[16, 17, 174] Peter Checkland and Jim Scholes (1999). “Soft Systems Methodology in Action.” New York, John Wiley and Sons.

## **Thesis**

[38, 67, 74, 162, 163, 164] N. Badr (2003) “An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software.” PhD Thesis, Liverpool John Moores University, UK.

## **Conference and Journal Papers**

[86] M. Adler, B. Davis, R. Weihmayer and W. Worrest (1989). “Conflict Resolution Strategies for Nonhierarchical Distributed Agents.” Proceeding of In Distributed Artificial Intelligence, volume 2, Morgan Kaufman Publishers Inc, San Francisco.

[147] L. Andrade and J. Fiadeiro (2002). “An Architectural Approach to Auto-Adaptive Systems.” Proceeding of ICDCS’02.

[10] Stephen Andriole and Leonard Adelman (1995). “Cognitive Systems Engineering for User-Computer Interface Design, Prototyping, and Evaluation.” New Jersey, Lawrence Elrbaum Associates.

[163, 154] J. Appavoo, B. Gamsa, G. Ganger, P. McKenney, M. Ostrowski, B. Rosenberg, M. Sturm and J. Xenidis (2003). “Enabling Autonomic Behaviour in Systems Software with Hot Swapping.” IBM Systems Journal.

[32] W. Ashby (1964). “An Introduction to Cybernetics.” London, Chapman and Hall.



- [36] D. Avison and G. Fitzgerald (1996). "Information Systems Development: Methodologies, Techniques and Tools." Maidenhead, McGraw-Hill Higher Education, 4<sup>th</sup> edition.
- [46] Baclawski, K. (2000). "Overview – Ontology and Abstract Model for Real-Time Distributed Resource Management." Self Controlling Software Group, Northeastern University.
- [128] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel and M. Spiteri (2000). "Generic Support for Distributed Applications." IEEE Computer Society Press, volume 33(3), pp68-76.
- [125] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajao, R. Strom and D. Sturman (1999). "An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems." Proceeding of ICDCS'99.
- [119] K. Barber, T. Liu, H. Goel and C. Martin (1999). "Conflict Representation and Classification in a Domain Independent Conflict Management Framework." Proceeding of the Third International Conference on Autonomous Agents.
- [90] K. Barber, T. Liu and D. Han (2000). "Strategy Selection-Based Meta-Level Reasoning." Proceeding of Agent Oriented Software Engineering, Limerick, Republic of Ireland.
- [89, 161] K.S. Barber, T. Liu and D.C. Han (2000). "Strategic Decision-Making for Conflict Resolution in Dynamic Organised Multi-Agent Systems." Proceeding of the International Conference on Distributed Systems Workshop, pp379-384.
- [123] M. Bearden, S. Garg and W. Lee (2001). "Integrated Goal Specification in Policy-Based Management." Proceeding of Policy 2001, International Workshop on Policies for Distributed Systems and Networks.
- [113] G. Bieber and L. Architect (1998). "Service-Oriented Programming", IBM Systems Journal, pp336-348. Reprinted in 2003.
- [70] G. Bieber and L. Architect (2003). "Service-Oriented Programming." Proceeding of IEEE International Conference on Services Computing, volume 1.
- [45] Bobrow, D.G., Gabriel, R.P and White, J.L. (1993). "CLOS in Context – The Shape of the Design Space." MIT Press, In A. Paepeke, ed., Object Oriented Programming – The CLOS Perspective, ch. 2.
- [100] M. Bratman (1987). "Intentions, Plans and Practical Reason." Harvard University Press 1987, pp35-41.
- [101, 102] M. Bratman, D.J. Israel and M. Pollack (1998). "Plans and Resource-Bounded Practical Reasoning" Computational Intelligence, volume 4(4).
- [42] W.L. Brogan (1991). "Modern Control Theory." New Jersey, Prentice Hall, 3<sup>rd</sup> edition.

- [178] R. Brooks (2006). "Conceptual Modelling." Proceeding of Operational Research Society Simulation Workshop SW'06.
- [122] M. Brunner and J. Quittek (2001). "MPLS Management Using Policies." Proceeding of IM2001: 2001 IEEE International Symposium on Integrated Network Management.
- [148] R. Buyya, S. Chapin and D. DiNucci (2000). "Architectural Models for Resource Management in the GRID." Proceeding of the First IEEE International Workshop on GRID Computing.
- [131] L. Cabrera, M. Jones and M. Theimer (2001). "Herald: Achieving a Global Event Notification Service." Proceeding of the Eighth Workshop on Hot Topics in OS'01.
- [126, 129] A. Carzaniga, J. Deng and A. Wolf (2001). "Fast Forwarding for Content-Based Networking." Proceeding of University of Colorado Report for Computer Science.
- [71] C. Castelfranchi (1996) "Conflict Ontology." Proceeding of ECAI 96 Workshop on Modelling Conflicts in Artificial Intelligence, Budapest.
- [104, 109] C. Castelfranchi, F. Dignum, C. Jonker and J. Truer (1999). "Deliberate Normative Agents: Principles and Architectures." Proceeding of ATAL99.
- [160] T. Chandra and S. Toueg (1996). "Unreliable Failure Detectors for Reliable Distributed Systems." Journal of the ACM, volume 43(2), pp225-267.
- [87] C. Chang, M. Chung and B. Lee (1994). "Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time". IEEE Transaction on Systems, volume 24.
- [144] S. Cheng, D. Gralan, B. Schmerl, J. Sousa, B. Spitzngel and P. Steenkiste (2002). "Using Architectural Style as a Basis for System Self-Repair." Third IEEE Conference on Software Architecture: System Design, Development and Maintenance, pp45-59.
- [150, 155] D. Chess, C. Palmer and S. White (2004). "Security in an Autonomic Computing Environment." IBM Systems Journal, 2004.
- [73] S. Cooper and A. Taleb-Bendiab (1997). "Consensus: Supporting Conflict Resolution Through Multi Parties Negotiation in Concurrent Engineering Design." Proceedings of 4<sup>th</sup> ISPE International Conference on Concurrent Engineering: Research and Application.
- [80, 83] S. Cooper and A. Taleb-Bendiab (1997). "A High Level Control mechanism for Managing Conflict Resolution in Concurrent Product Design." Proceeding of Fourth ISPE International Conference on Concurrent Engineering: Research and Application.



- [127] G. Cugola, E. Nitto and A. Fuggetta (1998). "The JEDI Event-Based Infrastructure and its Applications to the Development of the OPSS WFMS." IEEE Software Engineering, volume 27(9), pp827-850.
- [130] G. Cugola and E. Nitto (2001). "Using a Publish/Subscribe Middleware to Support Mobile Computing." Mobile Computing Workshop, Heidelberg, Germany.
- [98] M. Dastani, J. Hulstijn and L. van der Torre (2000). "BDI and QDT: a Comparison Based on Classical Decision Theory." Proceeding of AAAI Spring Symposium GTDT.
- [47] F-N Demers, and J. Malenfant (1995). "Reflection in Logic, Functional and Object-Oriented Programming: a Short Comparative Study". In Workshop of IJCAI'95 : On Reflection and Meta-Level Architecture and their Application in AI, pages 29-38, August 1995.
- [156] N. Diakov, H.J. Batteram, H. Zandbelt and M.J. Sinderen (2000). "Monitoring of Distributed Component Interactions." Proceeding of RM2000 Workshop on Reflective Middleware.
- [137] C. Efstratiou, A. Friday, N. Davies and K. Cheverst (2002). "A Platform Supporting Coordinated Adaptation in Mobile Systems." Proceeding of Fourth IEEE Workshop on Mobile Computing Systems and Applications.
- [81, 84] E. Ephrati and S. Rosenschein (1992). "The Clarke tax as a Consensus Mechanism Among Automated Agents." Proceeding of the Ninth Conference on Artificial Intelligence.
- [85] E. Ephrati and S. Rosenchein (1993). "Multi Agents Planning as a Dynamic Search for Social Consensus." 13<sup>th</sup> International Joint Conference on Artificial Intelligence.
- [145] Y. Eracar and M. Kokar (2000). "An Architecture for Software that Adapts to Changes in Requirements". Journal of Systems and Software, volume 50, pp209-219.
- [14, 107] R. Espejo and R. Harden (1989). "The Viable Systems Model. Interpretations and Applications of Stafford Beer's VSM". New York, John Wiley and Sons.
- [103, 108] J. Filipe (2002). "A Normative and Intentional Agent Model for Organisation Modelling." Proceeding of the Third International Workshop in Engineering Societies in the Agents World.
- [63] I. Foster, C. Kesselman, J. Nick and S. Tuecke (2002). "Grid Services for Distributed Systems Integration." IEEE Computer, 35 (6).
- [57, 58] David Garlan (2002). "Model Based Adaptation for Self-Healing Systems." ACM SIGSOFT Workshop on Self-Healing Systems, November 2002.

- [138] D. Garlan, S. Khersonsky and J. Soo Kim (2003). "Model Checking Publish-Subscribe Systems." Carnegie Mellon University Press.
- [143] D. Garlan and R. Stratton (2003). "Architecture-Based Adaptation of Complex Systems." Lecture Notes in Computer Science, DASADA, Software Engineering and Middleware, volume 2596/2003, pp157-173.
- [149] P. Horn (2001). "Autonomic Computing: IBM's Perspective on the State of Information Technology", IBM research paper, January 2001.
- [139] K. Jones (1993). "The Maturing of Software Architecture." Proceeding of Software Engineering Symposium Software Engineering Institute.
- [9, 190] Kinloch P. (2008). "GIS in Police Operations" Introduction to GIS seminar, Eastern Mediterranean University, GaziMagusa, Turkish Republic of Northern Cyprus, March 2008.
- [7, 188] Kinloch P. (2008). "GIS Reduces Crime" Intergraph SGI User Conference, Atkins, Epsom, UK, 1<sup>st</sup> - 2<sup>nd</sup> October 2008.
- [175] P. Kinloch and H. Francis (2006). "Delivering a New Information System Through the Use of SSM and VSM." First International Conference on GIS, Liverpool John Moores University, UK.
- [6, 187] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Police Operational Planning and Crime Detection Through Adopting an Information Systems Approach" Innovations 08, Al Ain University, United Arab Emirates, 16<sup>th</sup> - 18<sup>th</sup> December 2008.
- [8, 189] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Supporting Crime Detection and Operational Planning with Enhanced GIS" International Journal of Systems Research and Behavioural Science, January/February 2009, Vol 26 (1), pp 3-14, New York, John Wiley and Sons.
- [5, 186] Kinloch P., Francis H., Francis M., Taylor M. "Supporting Crime Detection and Operational Planning with Soft Systems Methodology and Viable Systems Model" CBS 2007, Karadeniz Technical University, Trabzon, Turkey, 29<sup>th</sup> October – 2<sup>nd</sup> November 2007.
- [77, 78] M. Klein (2000). "An Experiment Evaluation of Domain-Independent Fault Handling Services in Open Multi-Agent Systems." Proceeding of the International Conference on Multi-Agent Systems (ICMAS).
- [93] M. Klein "A Knowledge-Based Approach to Handling Exceptions in Workflow Systems," Journal of Computer-Supported Collaborative Work, 2000.
- [94, 95] M. Klein and C. Dellarocas (2000). "Towards a Systematic Repository of Knowledge About Managing Collaborative Design Conflicts." Proceeding of the International Conference on AI in Design, Boston.



- [91] M. Klein, J. Rodriguez-Aguliar and C. Dellarocas (2004). "Using Domain-Independent Exception Handling Services to Enable Robust Open Multi-Agent Systems." *The Case of Agent Death Journal for Autonomous Agents and Multi Agent Systems*, 2004.
- [55, 170] M. Kokar, K. Baslawski and Y. Eracar (1999). "Control and Theory-Based Foundation of Self-Controlling Software." *IEEE Intelligent Systems*, pp 37-45.
- [72] S. Lander and R. Lesser (1993). "Understanding the Role of Negotiation in Distributed Searches Among Heterogeneous Agents." *Proceeding of the Thirteenth International Joint Conference on Artificial Intelligence*.
- [76] S. Lander and V. Lesser (1992). "Customising Distributed Searches Among Agents with Heterogeneous Knowledge." *Proceeding of the First Conference on Information and Knowledge Management*.
- [4, 23, 24] A. Laws (2003). "CMSCB3001 Business Systems Analysis." *Liverpool John Moores University*.
- [40, 106] A. Laws, A. Taleb-Bendiab, S. Wade and D. Reilly (2003). "From Wetware to Software: A Cybernetic Approach". *Second International Workshop on Self-Adaptive Software, (IWSAS2001)*, 2001.
- [141] D. Le Metayer (2000). "Describing Software Engineering Styles Using Graph – Grammars." *IEEE Transactions on Software Engineering*, volume 24(7), pp521-533.
- [56] A. Ledeczi, A. Bakay and M. Maroti (2000). "Model-Integrated Embedded Systems." *First International Workshop on Self-Adaptive Software, (IWSAS2000)*, 2000.
- [88] V. Lesser and S. Lander (1991). "Customising Distributed Search Among Agents with Heterogeneous Knowledge." *Proceeding of 5<sup>th</sup> International Paper on AI Application in Manufacturing and Robotics*.
- [29] K. Lyytinen (1998). "Stakeholders, Information System Failures and Soft Systems Methodology: An Assessment." *Journal of Applied Systems Analysis*, volume 15, pp61-81.
- [50, 51] A. Meng (2000). "On Evaluation Self-Adaptive Software." *First International Workshop on Self-Adaptive Software, (IWSAS2000)*, 2000.
- [37] J. Mingers and A. Gill (2007). "Methodology – The Theory and Practice of Combining Management Science Methodologies." *New York, John Wiley and Sons*.
- [31, 34] A. Mirjamdotter (1998). "A Multi Model System Extension to Soft Systems Methodology." *Systems Research and Behavioural Science*, volume 18(2), pp181-196.
- [110] J. Moffett and M. Sloman "Policy Hierarchies for Distributed Systems Management". *IEEE Journal on Selected Areas in Communications*, 1993.

- [22] E. Mumford (1983). "Designing Human Systems, the ETHICS Approach." Manchester Business School.
- [11] R. Murray-Smith and T.A. Johansen (1997). "Multiple Model Approaches to Modelling and Control." *Control and Intelligent Systems*, volume 34(2), pp105-112.
- [68, 157, 158, 159] B. Nuseibeh, S. Easterbrook and A. Russo (2000). "Making Inconsistency Respectable in Software Development." *Journal of Systems and Software*, 56(11), pp171-180.
- [69] B. Nuseibeh, J. Kramer and A. Frinkelstein (1993). "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specifications." *IEEE Transactions on Software Engineering* 1994, pp187-196
- [97, 140] P. Oreizy and R. Taylor (1998). "On the Role of Software Architectures in Runtime System Reconfiguration." *Proceeding of the International Conference on Configurable Distributed Systems*, page 61.
- [146] P. Oriezy *et al.* (1999). "An Architecture-Based Approach to Self-Adaptive Software." *IEEE*, volume 14(3), pp54-62.
- [49, 165] L.J. Osterweil and L.A. Clarke (2000). "Continuous Self-Evaluation for the Self-Improvement of Software." *First International Workshop on Self-Adaptive Software*, 2000.
- [171] D. Pautler, S. Woods and A. Quilici (1997). "Exploring Domain-Specific Knowledge to Refine Simulation Specification." *Proceeding of Fifth IEEE International Symposium on Integrated Network Management*.
- [134] P. Pietzuch and J. Bacon (2002). "Hermes: A Distributed Event-Based Middleware Architecture." *Proceeding of the First International Workshop on Distributed Event-Based Systems*.
- [135] P. Pietzuch, B. Shand and J. Bacon (2003). "A Framework for Event Composition in Distributed Systems." *Proceeding of USENIX International Middleare Conference*.
- [99] A. Rao and M. Georgeff (1991). "Modelling Rational Agents Within a BDI-Architecture." *Proceeding of Knowledge Representation and Reasoning*, pp473-484.
- [172, 173] S. Rao and P. Georgeff (1995). "BDI Agents: From Theory to Practice." *The First International Conference on Multi-Agent Systems*.
- [59] P. Robertson (2000). "An Architecture for Self-Adaptive and its Application to Aerial Image Understanding" in *Proceeding of First International Workshop on Self-Adaptive Software, (IWSAS2000)*, 2000.
- [48] P. Robertson, R. Laddaga and H. Shrobe (2000). "Introduction: The First International Workshop on Self-Adaptive Software." *IWSA2000*.



- [79, 82] S. Rosenschein and G. Zlotkin (1994). "Rules of Encounter: Designing Conventions for Automated Negotiations Among Computers." MIT Press.
- [133] A. Rowstron, A. Kermarrec, M. Castro and P. Drusche (2001). "Scribe: The Design of a Large-Scale Event Notification Infrastructure." Proceeding of the Third International Workshop on Networked Group Communication.
- [136] M. Shanahan (1999). "The Event Calculus Explained". Artificial Intelligence Today, page 409.
- [60] I. Shaul (2000). "Dynamic Self-Adaptation in Distributed Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [53, 54] G. Simon, T. Kovacsazy and G. Peceli (2000). "Transient Management in Reconfigurable Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [111] M. Sloman (1994). "Policy Driven Management for Distributed Systems." Journal of Network and Systems Management, volume 2(4), pp333-360.
- [112] M. Sloman and E. Lupu (1999). "Policy Specification for Programmable Networks." Proceedings of First International Working Conference on Active Networks (IWAN99), Berlin.
- [114] M. Sloman, E. Lupu (1999). "Conflicts in Policy Based Distributed Systems Management." IEEE Transactions on Software Engineering.
- [115] M. Sloman and E. Lupu (1997). "Conflict Analysis for Management Policies." Proceeding of Fifth IEEE International Symposium on Integrated Network Management.
- [116] M. Sloman, E. Lupu, D. Marriott and N. Yialelis (1995). "A Policy Based Role Framework for Access Control." Proceeding of First ACM/NIST Workshop on Role-Based Access Control.
- [117] M. Sloman, E. Lupu, N. Damianou and N. Dulay (2001). "Ponder: A Language for Specifying Security and Management Policies for Distributed Systems." Imperial College Research Report.
- [118, 124] M. Sloman, E. Lupu and L. Lymberopoulos "An Adaptive Policy Based Management Framework for Differentiated Service Networks." Journal of Networks and Systems Management, volume 11(3), pp277-303.
- [75] K. Sycara (1991). "Cooperative Negotiation in Concurrent Engineering Design." Computer-Aided Cooperative Product Development, Springer Verlag Productions.
- [120] D. Verma and R. Jennings (2001). "Policy Based SLA, Management in Enterprise Networks." Proceeding of Policy 2001, International Workshop on Policies for Distributed Systems and Networks.

- [28] R. Veryard (1985). "What are Methodologies Good For?" *Data Processing*, volume 27(6), pp9-12.
- [92] A. Visser (1996) "An Exception-Handling Framework". *International Journal of Computer Integral Manufacturing*, volume 8(3), pp197-203.
- [35] I. von Bulow (1989). "The Bounding of a Problem Situation and the Concept of a System's Boundary in Soft Systems Methodology." *Journal of Applied Systems Analysis*, volume 16, pp35-41.
- [152] R. Want, T. Pening and D. Tennenhouse (2003). "Comparing Autonomic and Proactive Computing." *IBM Systems Journal*.
- [142] M. Wermelinger, A. Lopes and J. Fiadeiro (2001). "A Graph Based Architectural Configuration Language." *Proceeding of the Eighth European Software Engineering Conference*.
- [121] K. Yoshihara, M. Isomura and H. Horiuchi (2001). "Distributed Policy-Based Management Enabling Policy Adaptation on Monitoring Using Active Network Technology." *Proceeding of Twelfth IEEE International Workshop on Distributed Systems*.
- [132] S. Zhuang, B. Zhao and A. Joseph (2001). "Bayeux: An architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination." *Proceeding of the Eleventh International Workshop on network and OS Support for Digital Audio and Video*.

## Websites

- [52] ACME Website (2006). <http://www.cs.cmu.edu/~acme/> March 2007.
- [46] Baclawski, K. (2000). "Overview – Ontology and Abstract Model for Real-Time Distributed Resource Management." Self Controlling Software Group, Northeastern University.  
[http://www.ece.neu.edu/groups/scs/overview\\_documents/ontology.html](http://www.ece.neu.edu/groups/scs/overview_documents/ontology.html)
- [113] G. Bieber and L. Architect (2003). "Service-Oriented Programming." <http://www.openwings.org> and *Proceeding of IEEE International Conference on Services Computing*, volume 1.
- [39] DARPA Institute (2001) [http://www.darpa.mil/ito/Solicitations/PIP\\_9812.html](http://www.darpa.mil/ito/Solicitations/PIP_9812.html) December 2004.
- [47] F-N Demers, and J. Malenfant (1995). "Reflection in Logic, Functional and Object-Oriented Programming: a Short Comparative Study". In *Workshop of IJCAI'95 : On Reflection and Meta-Level Architecture and their Application in AI*, pages 29-38, August 1995. <http://citeseer.nj.nec.com/106401.html>



- [185] EtestingHub: The Testing Lifecycle  
[http://www.etestinghub.com/testing\\_lifecycles.php#2](http://www.etestinghub.com/testing_lifecycles.php#2) February 2007.
- [166, 167] R. Espejo (2003). "The Viable System Model: A Briefing About Organisational Structure." [www.syncho.com](http://www.syncho.com)
- [1, 41, 43, 44, 61, 62, 64, 65, 66, 96, 105, 149, 151, 191, 192] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [169] S. Krishana, P. Wagstrom and G. Laszewski (2003). "GSFL: A Workflow Framework for Grid Services." [www.cs.indiana.edu/~srikrish/talks/gsfl.ppt](http://www.cs.indiana.edu/~srikrish/talks/gsfl.ppt)
- [168] Liverpool John Moores University Enrich Project, [www.cms.livjm.ac.uk/enrich](http://www.cms.livjm.ac.uk/enrich) June 2006.
- [179, 180, 181, 182, 183] Merseyside Police Blueprint for Service Improvement, <http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf> September 2004.
- [184] National Intelligence Model  
<http://www.acpo.police.uk/asp/policies/Data/nim2005.pdf> January 2005.
- [2, 13, 20, 21] OGC, Prince2 official homepage. <http://www.ogc.gov.uk/prince2/> April 2006
- [176] Operational Research Society. <http://www.orsoc.org.uk/> 2006.
- [26] SmartDraw, SSADM software developers. <http://www.smartdraw.com> June 2006.
- [27] SSADM software developers Emerald Insight. <http://www.emeraldinsight.com> May 2008.
- [25] University of Salford lecture presentation.  
<http://www-users.cs.york.ac.uk/~kimble/nukais/twh.pdf> July 2008
- [177] "Viable Systems Model Guide."  
[http://www.esrad.org.uk/resources/vsmg\\_3/screen.php?page=lqguide](http://www.esrad.org.uk/resources/vsmg_3/screen.php?page=lqguide) August 2006.

## **References**

---

- [1] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [2] OGC, Prince2 official homepage. <http://www.ogc.gov.uk/prince2/> April 2006
- [3] Peter Checkland (1981). "Systems Thinking, Systems Practice." New York, John Wiley and Sons.
- [4] A. Laws (2003). "CMSCB3001 Business Systems Analysis." Liverpool John Moores University.
- [5] Kinloch P., Francis H., Francis M., Taylor M. "Supporting Crime Detection and Operational Planning with Soft Systems Methodology and Viable Systems Model" CBS 2007, Karadeniz Technical University, Trabzon, Turkey, 29<sup>th</sup> October – 2<sup>nd</sup> November 2007.
- [6] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Police Operational Planning and Crime Detection Through Adopting an Information Systems Approach" Innovations 08, Al Ain University, United Arab Emirates, 16<sup>th</sup> - 18<sup>th</sup> December 2008.
- [7] Kinloch P. (2008). "GIS Reduces Crime" Intergraph SGI User Conference, Atkins, Epsom, UK, 1<sup>st</sup> - 2<sup>nd</sup> October 2008.
- [8] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Supporting Crime Detection and Operational Planning with Enhanced GIS" International Journal of Systems Research and Behavioural Science, January/February 2009, Vol 26 (1), pp 3-14, New York, John Wiley and Sons.
- [9] Kinloch P. (2008). "GIS in Police Operations" Introduction to GIS seminar, Eastern Mediterranean University, GaziMagusa, Turkish Republic of Northern Cyprus, March 2008.
- [10] Stephen Andriole and Leonard Adelman (1995). "Cognitive Systems Engineering for User-Computer Interface Design, Prototyping, and Evaluation." New Jersey, Lawrence Elrbaum Associates.
- [11] R. Murray-Smith and T.A. Johansen (1997). "Multiple Model Approaches to Modelling and Control." Control and Intelligent Systems, volume 34(2), pp105-112.
- [12] Peter Checkland (1981). "Systems Thinking, Systems Practice." New York, John Wiley and Sons.
- [13] OGC, Prince2 official homepage. <http://www.ogc.gov.uk/prince2/> April 2006.
- [14] R. Espejo and R. Harden (1989). "The Viable Systems Model. Interpretations and Applications of Stafford Beer's VSM". New York, John Wiley and Sons.
- [15] Peter Checkland and Sue Holwell (1998). "Information, Systems and Information Systems." New York, John Wiley and Sons.
- [16] Peter Checkland and Jim Scholes (1999). "Soft Systems Methodology in Action." New York, John Wiley and Sons.
- [17] Peter Checkland and Jim Scholes (1999). "Soft Systems Methodology in Action." New York, John Wiley and Sons.
- [18] S. Beer (1994). "Diagnosing the System for Organisations." New York, John Wiley & Sons, 1994 version.
- [19] S. Beer (1994). "Diagnosing the System for Organisations." New York, John Wiley & Sons, 1994 version.
- [20] OGC, Prince2 official homepage. <http://www.ogc.gov.uk/prince2/> April 2006
- [21] OGC, Prince2 official homepage. <http://www.ogc.gov.uk/prince2/> April 2006
- [22] E. Mumford (1983). "Designing Human Systems, the ETHICS Approach." Manchester Business School.



- 
- [23] A. Laws (2003). "CMSCB3001 Business Systems Analysis." Liverpool John Moores University.
- [24] A. Laws (2003). "CMSCB3001 Business Systems Analysis." Liverpool John Moores University.
- [25] University of Salford lecture presentation. <http://www-users.cs.york.ac.uk/~kimble/nukais/twh.pdf> July 2008
- [26] SmartDraw, SSADM software developers. <http://www.smartdraw.com> June 2006.
- [27] SSADM software developers Emerald Insight. <http://www.emeraldinsight.com> May 2008.
- [28] R. Veryard (1985). "What are Methodologies Good For?" Data Processing, volume 27(6), pp9-12.
- [29] K. Lyytinen (1998). "Stakeholders, Information System Failures and Soft Systems Methodology: An Assessment." Journal of Applied Systems Analysis, volume 15, pp61-81.
- [30] Peter Checkland (1981). "Systems Thinking, Systems Practice." New York, John Wiley and Sons.
- [31] A. Mirijamdotter (1998). "A Multi Model System Extension to Soft Systems Methodology." Systems Research and Behavioural Science, volume 18(2), pp181-196.
- [32] W. Ashby (1964). "An Introduction to Cybernetics." London, Chapman and Hall.
- [33] S. Beer (1994). "Diagnosing the System for Organisations." New York, John Wiley & Sons, 1994 version.
- [34] A. Mirijamdotter (1998). "A Multi Model System Extension to Soft Systems Methodology." Systems Research and Behavioural Science, volume 18(2), pp181-196.
- [35] I. von Bulow (1989). "The Bounding of a Problem Situation and the Concept of a System's Boundary in Soft Systems Methodology." Journal of Applied Systems Analysis, volume 16, pp35-41.
- [36] D Avison and G. Fitzgerald (1996). "Information Systems Development: Methodologies, Techniques and Tools." Maidenhead, McGraw-Hill Higher Education, 4th edition.
- [37] J. Mingers and A. Gill (2007). "Methodology – The Theory and Practice of Combining Management Science Methodologies." New York, John Wiley and Sons.
- [38] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, Liverpool John Moores University, UK.
- [39] DARPA Institute (2001) [http://www.darpa.mil/ito/Solicitations/PIP\\_9812.html](http://www.darpa.mil/ito/Solicitations/PIP_9812.html) December 2004.
- [40] A. Laws, A. Taleb-Bendiab, S. Wade and D. Reilly (2003). "From Wetware to Software: A Cybernetic Approach". Second International Workshop on Self-Adaptive Software, (IWSAS2001), 2001.
- [41] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [42] W.L. Brogan (1991). "Modern Control Theory." New Jersey, Prentice Hall, 3rd edition.
- [43] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [44] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006



- 
- [45] Bobrow, D.G., Gabriel, R.P and White, J.L. (1993). "CLOS in Context – The Shape of the Design Space." MIT Press, In A. Paepeke, ed., Object Oriented Programming – The CLOS Perspective, ch. 2.
- [46] Baclawski, K. (2000). "Overview – Ontology and Abstract Model for Real-Time Distributed Resource Management." Self Controlling Software Group, Northeastern University.  
[http://www.ece.neu.edu/groups/scs/overview\\_documents/ontology.html](http://www.ece.neu.edu/groups/scs/overview_documents/ontology.html)
- [47] F-N Demers, and J. Malenfant (1995). "Reflection in Logic, Functional and Object-Oriented Programming: a Short Comparative Study". In Workshop of IJCAI'95 : On Reflection and Meta-Level Architecture and their Application in AI, pages 29-38, August 1995. <http://citeseer.nj.nec.com/106401.html>
- [48] P. Robertson, R. Laddaga and H. Shrobe (2000). "Introduction: The First International Workshop on Self-Adaptive Software." IWSA2000.
- [49] L.J. Osterweil and L.A. Clarke (2000). "Continuous Self-Evaluation for the Self-Improvement of Software." First International Workshop on Self-Adaptive Software, 2000.
- [50] A. Meng (2000). "On Evaluation Self-Adaptive Software." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [51] A. Meng (2000). "On Evaluation Self-Adaptive Software." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [52] ACME Website (2006). <http://www.cs.cmu.edu/~acme/> March 2007.
- [53] G. Simon, T. Kovacsazy and G. Peceli (2000). "Transient Management in Reconfigurable Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [54] G. Simon, T. Kovacsazy and G. Peceli (2000). "Transient Management in Reconfigurable Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [55] M. Kokar, K. Baslawski and Y. Eracar (1999). "Control and Theory-Based Foundation of Self-Controlling Software." IEEE Intelligent Systems, pp 37-45.
- [56] A. Ledeczi, A. Bakay and M. Maroti (2000). "Model-Integrated Embedded Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [57] David Garlan (2002). "Model Based Adaptation for Self-Healing Systems." ACM SIGSOFT Workshop on Self-Healing Systems, November 2002.
- [58] David Garlan (2002). "Model Based Adaptation for Self-Healing Systems." ACM SIGSOFT Workshop on Self-Healing Systems, November 2002.
- [59] P. Robertson (2000). "An Architecture for Self-Adaptive and its Application to Aerial Image Understanding" in Proceeding of First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [60] I. Shaul (2000). "Dynamic Self-Adaptation in Distributed Systems." First International Workshop on Self-Adaptive Software, (IWSAS2000), 2000.
- [61] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [62] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [63] I. Foster, C. Kesselman, J. Nick and S. Tuecke (2002). "Grid Services for Distributed Systems Integration." IEEE Computer, 35 (6).
- [64] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [65] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [66] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006



- 
- [67] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, Liverpool John Moores University, UK.
- [68] B. Nuseibeh, S. Easterbrook and A. Russo (2000). "Making Inconsistency Respectable in Software Development." *Journal of Systems and Software*, 56(11), pp171-180.
- [69] B. Nuseibeh, J. Kramer and A. Frinkelstein (1993). "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specifications." *IEEE Transactions on Software Engineering* 1994, pp187-196
- [70] G. Bieber and L. Architect (1998). "Service-Oriented Programming", *IBM Systems Journal*, pp336-348. Reprinted in 2003.
- [71] C. Castelfranchi (1996) "Conflict Ontology." *Proceeding of ECAI 96 Workshop on Modelling Conflicts in Artificial Intelligence*, Budapest.
- [72] S. Lander and R. Lesser (1993). "Understanding the Role of Negotiation in Distributed Searches Among Heterogeneous Agents." *Proceeding of the Thirteenth International Joint Conference on Artificial Intelligence*.
- [73] S. Cooper and A. Taleb-Bendiab (1997). "Consensus: Supporting Conflict Resolution Through Multi Parties Negotiation in Concurrent Engineering Design." *Proceedings of 4th ISPE International Conference on Concurrent Engineering: Research and Application*.
- [74] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, Liverpool John Moores University, UK.
- [75] K. Sycara (1991). "Cooperative Negotiation in Concurrent Engineering Design." *Computer-Aided Cooperative Product Development*, Springer Verlag Productions.
- [76] S. Lander and V. Lesser (1992). "Customising Distributed Searches Among Agents with Heterogeneous Knowledge." *Proceeding of the First Conference on Information and Knowledge Management*.
- [77] M. Klein (2000). "An Experiment Evaluation of Domain-Independent Fault Handling Services in Open Multi-Agent Systems." *Proceeding of the International Conference on Multi-Agent Systems (ICMAS)*.
- [78] M. Klein (2000). "An Experiment Evaluation of Domain-Independent Fault Handling Services in Open Multi-Agent Systems." *Proceeding of the International Conference on Multi-Agent Systems (ICMAS)*.
- [79] S. Rosenschein and G. Zlotkin (1994). "Rules of Encounter: Designing Conventions for Automated Negotiations Among Computers." MIT Press.
- [80] S. Cooper and A. Taleb-Bendiab (1997). "A High Level Control mechanism for Managing Conflict Resolution in Concurrent Product Design." *Proceeding of Fourth ISPE International Conference on Concurrent Engineering: Research and Application*.
- [81] E. Ephrati and S. Rosenschein (1992). "The Clarke tax as a Consensus Mechanism Among Automated Agents." *Proceeding of the Ninth Conference on Artificial Intelligence*.
- [82] J. Rosenschein and G. Zlotkin (1994). "Rules of Encounter: Designing Conventions for Automated Negotiations Among Computers." MIT Press.
- [83] S. Cooper and A. Taleb-Bendiab (1997). "A High Level Control mechanism for Managing Conflict Resolution in Concurrent Product Design." *Proceeding of Fourth ISPE International Conference on Concurrent Engineering: Research and Application*.



- 
- [84] E. Ephrati and S. Rosenschein (1992). "The Clarke tax as a Consensus Mechanism Among Automated Agents." Proceeding of the Ninth Conference on Artificial Intelligence.
- [85] E. Ephrati and S. Rosenchein (1993). "Multi Agents Planning as a Dynamic Search for Social Consensus." 13th International Joint Conference on Artificial Intelligence.
- [86] M. Adler, B. Davis, R. Weihmayer and W. Worrest (1989). "Conflict Resolution Strategies for Nonhierarchical Distributed Agents." Proceeding of In Distributed Artificial Intelligence, volume 2, Morgan Kaufman Publishers Inc, San Francisco.
- [87] C. Chang, M. Chung and B. Lee (1994). "Collision Avoidance of Two General Robot Manipulators by Minimum Delay Time". IEEE Transaction on Systems, volume 24.
- [88] V. Lesser and S. Lander (1991). "Customising Distributed Search Among Agents with Heterogeneous Knowledge." Proceeding of 5th International Paper on AI Application in Manufacturing and Robotics.
- [89] K. Barber, T. Liu and D. Han (2000). "Strategic Decision-Making for Conflict Resolution in Dynamic Organised Multi-Agent Systems." Proceeding of GDN 2000.
- [90] K. Barber, T. Liu and D. Han (2000). "Strategy Selection-Based Meta-Level Reasoning." Proceeding of Agent Oriented Software Engineering, Limerick, Republic of Ireland.
- [91] M. Klein, J. Rodriguez-Aguliar and C. Dellarocas (2004). "Using Domain-Independent Exception Handling Services to Enable Robust Open Multi-Agent Systems." The Case of Agent Death Journal for Autonomous Agents and Multi Agent Systems, 2004.
- [92] A. Visser (1996) "An Exception-Handling Framework". International Journal of Computer Integral Manufacturing, volume 8(3), pp197-203.
- [93] M. Klein "A Knowledge-Based Approach to Handling Exceptions in Workflow Systems", Journal of Computer-Supported Collaborative Work, 2000.
- [94] M. Klein and C. Dellarocas (2000). "Towards a Systematic Repository of Knowledge About Managing Collaborative Design Conflicts." Proceeding of the International Conference on AI in Design, Boston.
- [95] M. Klein and C. Dellarocas (2000). "Towards a Systematic Repository of Knowledge About Managing Collaborative Design Conflicts." Proceeding of the International Conference on AI in Design, Boston.
- [96] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [97] P. Oreizy and R. Taylor (1998). "On the Role of Software Architectures in Runtime System Reconfiguration." Proceedings of the International Conference on Configurable Distributed Systems.
- [98] M. Dastani, J. Hulstijn and L. van der Torre (2000). "BDI and QDT: a Comparison Based on Classical Decision Theory." Proceeding of AAAI Spring Symposium GTDT.
- [99] A. Rao and M. Georgeff (1991). "Modelling Rational Agents Within a BDI-Architecture." Proceeding of Knowledge Representation and Reasoning, pp473-484.
- [100] M. Bratman (1987). "Intentions, Plans and Practical Reason." Harvard University Press 1987, pp35-41.
- [101] M. Bratman, D.J. Israel and M. Pollack (1998). "Plans and Resource-Bounded Practical Reasoning" Computational Intelligence, volume 4(4).
- [102] M. Bratman, D.J. Israel and M. Pollack (1998). "Plans and Resource-Bounded Practical Reasoning" Computational Intelligence, volume 4(4).



- 
- [103] J. Filipe (2002). "A Normative and Intentional Agent Model for Organisation Modelling." Proceeding of the Third International Workshop in Engineering Societies in the Agents World.
- [104] C. Castelfranchi, F. Dignum, C. Jonker and J. Truer (1999). "Deliberate Normative Agents: Principles and Architectures." Proceeding of ATAL99.
- [105] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [106] A. Laws, A. Taleb-Bendiab, S. Wade and D. Reilly (2003). "From Wetware to Software: A Cybernetic Approach". Second International Workshop on Self-Adaptive Software, (IWSAS2001), 2001.
- [107] R. Espejo and R. Harden (1989). "The Viable Systems Model. Interpretations and Applications of Stafford Beer's VSM". New York, John Wiley and Sons.
- [108] J. Filipe (2002). "A Normative and Intentional Agent Model for Organisation Modelling." Proceeding of the Third International Workshop in Engineering Societies in the Agents World.
- [109] C. Castelfranchi, F. Dignum, C. Jonker and J. Truer (1999). "Deliberate Normative Agents: Principles and Architectures." Proceeding of ATAL99.
- [110] J. Moffett and M. Sloman "Policy Hierarchies for Distributed Systems Management". IEEE Journal on Selected Areas in Communications, 1993.
- [111] M. Sloman (1994). "Policy Driven Management for Distributed Systems." Journal of Network and Systems Management, volume 2(4), pp333-360.
- [112] M. Sloman and E. Lupu (1999). "Policy Specification for Programmable Networks." Proceedings of First International Working Conference on Active Networks (IWAN99), Berlin.
- [113] G. Bieber and L. Architect (2003). "Service-Oriented Programming." <http://www.openwings.org> and Proceeding of IEEE International Conference on Services Computing, volume 1.
- [114] M. Sloman, E. Lupu (1999). "Conflicts in Policy Based Distributed Systems Management." IEEE Transactions on Software Engineering.
- [115] M. Sloman and E. Lupu (1997). "Conflict Analysis for Management Policies." Proceeding of Fifth IEEE International Symposium on Integrated Network Management.
- [116] M. Sloman, E. Lupu, D. Marriott and N. Yiaelis (1995). "A Policy Based Role Framework for Access Control." Proceeding of First ACM/NIST Workshop on Role-Based Access Control.
- [117] M. Sloman, E. Lupu, N. Damianou and N. Dulay (2001). "Ponder: A Language for Specifying Security and Management Policies for Distributed Systems." Imperial College Research Report.
- [118] M. Sloman, E. Lupu and L. Lymberopoulos "An Adaptive Policy Based Management Framework for Differentiated Service Networks." Journal of Networks and Systems Management, volume 11(3), pp277-303.
- [119] K. Barber, T. Liu, H. Goel and C. Martin (1999). "Conflict Representation and Classification in a Domain Independent Conflict Management Framework." Proceeding of the Third International Conference on Autonomous Agents.
- [120] D. Verma and R. Jennings (2001). "Policy Based SLA, Management in Enterprise Networks." Proceeding of Policy 2001, International Workshop on Policies for Distributed Systems and Networks.
- [121] K. Yoshihara, M. Isomura and H. Horiuchi (2001). "Distributed Policy-Based Management Enabling Policy Adaptation on Monitoring Using Active Network



- 
- Technology.” Proceeding of Twelfth IEEE International Workshop on Distributed Systems.
- [122] M. Brunner and J. Quittek (2001). “MPLS Management Using Policies.” Proceeding of IM2001: 2001 IEEE International Symposium on Integrated Network Management.
- [123] M. Bearden, S. Garg and W. Lee (2001). “Integrated Goal Specification in Policy-Based Management.” Proceeding of Policy 2001, International Workshop on Policies for Distributed Systems and Networks.
- [124] M. Sloman, E. Lupu and L. Lymberopoulos “An Adaptive Policy Based Management Framework for Differentiated Service Networks.” *Journal of Networks and Systems Management*, volume 11(3), pp277-303.
- [125] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajao, R. Strom and D. Sturman (1999). “An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems.” Proceeding of ICDCS’99.
- [126] A. Carzaniga, J. Deng and A. Wolf (2001). “Fast Forwarding for Content-Based Networking.” Proceeding of University of Colorado Report for Computer Science.
- [127] G. Cugola, E. Nitto and A. Fuggetta (1998). “The JEDI Event-Based Infrastructure and its Applications to the Development of the OPSS WFMS.” *IEEE Software Engineering*, volume 27(9), pp827-850.
- [128] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel and M. Spiteri (2000). “Generic Support for Distributed Applications.” *IEEE Computer Society Press*, volume 33(3), pp68-76.
- [129] A. Carzaniga, J. Deng and A. Wolf (2001). “Fast Forwarding for Content-Based Networking.” Proceeding of University of Colorado Report for Computer Science.
- [130] G. Cugola and E. Nitto (2001). “Using a Publish/Subscribe Middleware to Support Mobile Computing.” *Mobile Computing Workshop*, Heidelberg, Germany.
- [131] L. Cabrera, M. Jones and M. Theimer (2001). “Herald: Achieving a Global Event Notification Service.” Proceeding of the Eighth Workshop on Hot Topics in OS’01.
- [132] S. Zhuang, B. Zhao and A. Joseph (2001). “Bayeux: An architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination.” Proceeding of the Eleventh International Workshop on network and OS Support for Digital Audio and Video.
- [133] A. Rowstron, A. Kermarrec, M. Castro and P. Drusche (2001). “Scribe: The Design of a Large-Scale Event Notification Infrastructure.” Proceeding of the Third International Workshop on Networked Group Communication.
- [134] P. Pietzuch and J. Bacon (2002). “Hermes: A Distributed Event-Based Middleware Architecture.” Proceeding of the First International Workshop on Distributed Event-Based Systems.
- [135] P. Pietzuch, B. Shand and J. Bacon (2003). “A Framework for Event Composition in Distributed Systems.” Proceeding of USENIX International Middleare Conference.
- [136] M. Shanahan (1999). “The Event Calculus Explained”. *Artificial Intelligence Today*, page 409.
- [137] C. Efstratiou, A. Friday, N. Davies and K. Cheverst (2002). “A Platform Supporting Coordinated Adaptation in Mobile Systems.” Proceeding of Fourth IEEE Workshop on Mobile Computing Systems and Applications.



- 
- [138] D. Garlan, S. Khersonsky and J. Soo Kim (2003). "Model Checking Publish-Subscribe Systems." Carnegie Mellon University Press.
- [139] K. Jones (1993). "The Maturing of Software Architecture." Proceeding of Software Engineering Symposium Software Engineering Institute.
- [140] P. Oreizy and R. Taylor (1998). "On the Role of Software Architectures in Runtime System Reconfiguration." Proceeding of the International Conference on Configurable Distributed Systems, page 61.
- [141] D. Le Metayer (2000). "Describing Software Engineering Styles Using Graph – Grammars." IEEE Transactions on Software Engineering, volume 24(7), pp521-533.
- [142] M. Wermelinger, A. Lopes and J. Fiadeiro (2001). "A Graph Based Architectural Configuration Language." Proceeding of the Eighth European Software Engineering Conference.
- [143] D. Garlan and R. Stratton (2003). "Architecture-Based Adaptation of Complex Systems." Lecture Notes in Computer Science, DASADA, Software Engineering and Middleware, volume 2596/2003, pp157-173.
- [144] S. Cheng, D. Gralan, B. Schmerl, J. Sousa, B. Spitzngel and P. Steenkiste (2002). "Using Architectural Style as a Basis for System Self-Repair." Third IEEE Conference on Software Architecture: System Design, Development and Maintenance, pp45-59.
- [145] Y. Eracar and M. Kokar (2000). "An Architecture for Software that Adapts to Changes in Requirements". Journal of Systems and Software, volume 50, pp209-219.
- [146] P. Oriezy *et al.* (1999). "An Architecture-Based Approach to Self-Adaptive Software." IEEE, volume 14(3), pp54-62.
- [147] L. Andrade and J. Fiadeiro (2002). "An Architectural Approach to Auto-Adaptive Systems." Proceeding of ICDCS'02.
- [148] R. Buyya, S. Chapin and D. DiNucci (2000). "Architectural Models for Resource Management in the GRID." Proceeding of the First IEEE International Workshop on GRID Computing.
- [149] P. Horn (2001). "Autonomic Computing: IBM's Perspective on the State of Information Technology", IBM research paper at <http://www.ibm.com/autonomic> January 2001.
- [150] D. Chess, C. Palmer and S. White (2004). "Security in an Autonomic Computing Environment." IBM Systems Journal, 2004.
- [151] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [152] R. Want, T. Pening and D. Tennenhouse (2003). "Comparing Autonomic and Proactive Computing." IBM Systems Journal.
- [153] J. Appavoo, B. Gamsa, G. Ganger, P. McKenney, M. Ostrowski, B. Rosenberg, M. Sturm and J. Xenidis (2003). "Enabling Autonomic Behaviour in Systems Software with Hot Swapping." IBM Systems Journal.
- [154] J. Appavoo, B. Gamsa, G. Ganger, P. McKenney, M. Ostrowski, B. Rosenberg, M. Sturm and J. Xenidis (2003). "Enabling Autonomic Behaviour in Systems Software with Hot Swapping." IBM Systems Journal.
- [155] D. Chess, C. Palmer and S. White (2004). "Security in an Autonomic Computing Environment." IBM Systems Journal, 2004.
- [156] N. Diakov, H.J. Batteram, H. Zandbelt and M.J. Sinderen (2000). "Monitoring of Distributed Component Interactions." Proceeding of RM2000 Workshop on Reflective Middleware.



- 
- [157] B. Nuseibeh, S. Easterbrook and A. Russo (2000). "Making Inconsistency Respectable in Software Development." *Journal of Systems and Software*, 56(11), pp171-180.
- [158] B. Nuseibeh, S. Easterbrook and A. Russo (2000). "Making Inconsistency Respectable in Software Development." *Journal of Systems and Software*, 56(11), pp171-180.
- [159] B. Nuseibeh, S. Easterbrook and A. Russo (2000). "Making Inconsistency Respectable in Software Development." *Journal of Systems and Software*, 56(11), pp171-180.
- [160] T. Chandra and S. Toueg (1996). "Unreliable Failure Detectors for Reliable Distributed Systems." *Journal of the ACM*, volume 43(2), pp225-267.
- [161] K.S. Barber, T.H.H. Liu and D.C. Han (2000). "Strategic Decision-Making for Conflict Resolution in Dynamic Organised Multi-Agent Systems." *Proceeding of the International Conference on Distributed Systems Workshop*, pp379-384.
- [162] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, Liverpool John Moores University, UK.
- [163] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, *Liverpool John Moores University*, UK.
- [164] N. Badr (2003) "An Investigation into Autonomic Middleware Control Services to Support Distributed Self-Adaptive Software." PhD Thesis, Liverpool John Moores University, UK.
- [165] L.J. Osterweil and L.A. Clarke (2000). "Continuous Self-Evaluation for the Self-Improvement of Software." *First International Workshop on Self-Adaptive Software*.
- [166] R. Espejo (2003). "The Viable System Model: A Briefing About Organisational Structure." [www.syncho.com](http://www.syncho.com)
- [167] R. Espejo (2003). "The Viable System Model: A Briefing About Organisational Structure." [www.syncho.com](http://www.syncho.com)
- [168] [www.cms.livjm.ac.uk/enrich](http://www.cms.livjm.ac.uk/enrich) June 20006.
- [169] S. Krishana, P. Wagstrom and G. Laszewski (2003). "GSFL: A Workflow Framework for Grid Services." [www.cs.indiana.edu/~srikrish/talks/gsfl.ppt](http://www.cs.indiana.edu/~srikrish/talks/gsfl.ppt)
- [170] M. Kokar, K. Baslawski and Y. Eracar (1999). "Control and Theory-Based Foundation of Self-Controlling Software." *IEEE Intelligent Systems*, pp 37-45.
- [171] D. Pautler, S. Woods and A. Quilici (1997). "Exploring Domain-Specific Knowledge to Refine Simulation Specification." *Proceeding of Fifth IEEE International Symposium on Integrated Network Management*.
- [172] S. Rao and P. Georgeff (1995). "BDI Agents: From Theory to Practice." *The First International Conference on Multi-Agent Systems*.
- [173] S. Rao and P. Georgeff (1995). "BDI Agents: From Theory to Practice." *The First International Conference on Multi-Agent Systems*.
- [174] Peter Checkland and Jim Scholes (1999). "Soft Systems Methodology in Action." New York, John Wiley and Sons.
- [175] P. Kinloch and H. Francis (2006). "Delivering a New Information System Through the Use of SSM and VSM." *First International Conference on GIS*, Liverpool John Moores University, UK.
- [176] Operational Research Society. <http://www.orsoc.org.uk/> 2006.



- 
- [177] "Viable Systems Model Guide."  
[http://www.esrad.org.uk/resources/vsmg\\_3/screen.php?page=lqguide](http://www.esrad.org.uk/resources/vsmg_3/screen.php?page=lqguide) August 2006.
- [178] R. Brooks (2006). "Conceptual Modelling." Proceeding of Operational Research Society Simulation Workshop SW'06.
- [179] Merseyside Police Blueprint for Service Improvement,  
<http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf>  
September 2004.
- [180] Merseyside Police Blueprint for Service Improvement,  
<http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf>  
September 2004.
- [181] Merseyside Police Blueprint for Service Improvement,  
<http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf>  
September 2004.
- [182] Merseyside Police Blueprint for Service Improvement,  
<http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf>  
September 2004.
- [183] Merseyside Police Blueprint for Service Improvement,  
<http://www.merseysidepoliceauthority.gov.uk/resources/181264/blueprint1.pdf>  
September 2004.
- [184] National Intelligence Model  
<http://www.acpo.police.uk/asp/policies/Data/nim2005.pdf> January 2005.
- [185] EtestingHub: The Testing Lifecycle  
[http://www.etestinghub.com/testing\\_lifecycles.php#2](http://www.etestinghub.com/testing_lifecycles.php#2) February 2007.
- [186] Kinloch P., Francis H., Francis M., Taylor M. "Supporting Crime Detection and Operational Planning with Soft Systems Methodology and Viable Systems Model" CBS 2007, Karadeniz Technical University, Trabzon, Turkey, 29<sup>th</sup> October – 2<sup>nd</sup> November 2007.
- [187] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Police Operational Planning and Crime Detection Through Adopting an Information Systems Approach" Innovations 08, Al Ain University, United Arab Emirates, 16<sup>th</sup> - 18<sup>th</sup> December 2008.
- [188] Kinloch P. (2008). "GIS Reduces Crime" Intergraph SGI User Conference, Atkins, Epsom, UK, 1<sup>st</sup> - 2<sup>nd</sup> October 2008.
- [189] Kinloch P., Francis H., Francis M., Taylor M. (2008). "Supporting Crime Detection and Operational Planning with Enhanced GIS" International Journal of Systems Research and Behavioural Science, January/February 2009, Vol 26 (1), pp 3-14, New York, John Wiley and Sons.
- [190] Kinloch P. (2008). "GIS in Police Operations" Introduction to GIS seminar, Eastern Mediterranean University, GaziMagusa, Turkish Republic of Northern Cyprus, March 2008.
- [191] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006
- [192] IBM autonomic research (2003). <http://www.ibm.com/autonomic> April 2006