CRANFIELD UNIVERSITY

MARK STELLING

Automated Cost and Customer Based Business Process Reengineering in the Service Sector

SCHOOL OF APPLIED SCIENCES
PhD Thesis

CRANFIELD UNIVERSITY

SCHOOL OF APPLIED SCIENCES

PHD THESIS

October 2008

MARK STELLING

Automated Cost and Customer Based Business Process Reengineering in the Service Sector

Supervisors: Prof. Rajkumar Roy and Dr. Ashutosh Tiwari

This thesis is submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

THIS PAGE IS INTENTIONALLY LEFT BLANK

Abstract

The design of business processes often ignores detailed consideration of service cost. With competitive market pressure, this has become a key factor for the service sector. Along with cost, customer satisfaction is a driving force in all organisations these days. In an increasingly competitive marketplace, due to the current emphasis on service throughout the whole economy, businesses must fight to attract and retain their customers. The response time for modification, or reengineering, of existing processes and the creation of new processes is important in the service sector – this is usually required in a short period of time in order to respond to market and/or customer demand. Thus, there is a requirement for the automation of business process reengineering in order to facilitate this and to minimise response time.

In order to address the above, the research carried out during this project involved the design and development of a framework to integrate an activity based cost estimating approach which takes into account probability of resource usage in variable processes, with an automated business process reengineering technique which incorporates an evolutionary computing (genetic algorithms) based optimisation module. Along with this, a novel methodology for detecting risk of negative impact on levels of customer satisfaction without the availability of customer related data has also been developed and integrated with the reengineering technique - cost reduction being the primary objective, but not at the expense of customer satisfaction. The overall aim is to automate the reengineering of business processes to as great an extent as possible in order to save potentially considerable human time and effort. The development of the automated framework also included the creation of a numeric process representation mechanism in order to enable quantitative analysis of complex business processes.

The framework is implemented within a prototype software platform for expert validation.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Acknowledgements

Firstly, I would like to express the utmost gratitude to my supervisors, Professor Rajkumar Roy and Doctor Ashutosh Tiwari, for their constant support and guidance during this PhD. Thanks also to Doctor Ip-Shing Fan, my assessor, for his valuable input during the first two years of the PhD.

I would also like to thank the funding body of this research, the Engineering and Physical Sciences Research Council, and its industrial collaborator, British Telecom - particularly my industrial supervisor at BT, Doctor Basim Majeed, for all his help and support, and Professor Ben Azvine for his contributions.

I would like to thank all the members of the Decision Engineering Centre (DEC) for the useful discussions we had throughout the past 3 years, and my many friends at Cranfield University for making my time there so enjoyable.

Finally, I would like to thank my Mum and Dad for their ongoing love, support and encouragement.

THIS PAGE IS INTENTIONALLY LEFT BLANK

List of Publications

Stelling, M.T., Roy, R., Tiwari, A., and Majeed, B. (2006), 'A Coding Mechanism for Business Process Optimisation Using Genetic Algorithms: Initial Thoughts', *10th IASTED International Conference on Artificial Intelligence and Soft Computing,* Palma, Majorca, ACTA Press, CA.

Stelling, M.T., Roy, R., Tiwari, A. (Publication date March 2009), 'A Novel Modelling & Optimisation Technique for Business Processes: An Evolutionary Computing Based Approach', *WSC12 online conference*, Springer, London.

Stelling, M.T., Roy, R., Majeed, B. and Tiwari, A. (Provisionally accepted Oct. 2008), 'Automated Business Process Reengineering: a Methodology for Detecting Risk of Negative Impact on Customer Satisfaction', *BT Innovation Journal.*

Stelling, M.T., Roy, R., Tiwari, A. and Majeed, B. (Submitted Sept. 2008), 'Development of an Automated Business Process Reengineering Approach', *Computers in Industry.*

Stelling, M.T., Roy, R., Tiwari, A. and Majeed, B. (Accepted for publication Dec. 2008), 'Evaluation of Business Processes using Probability Driven Activity Based Costing', *The Service Industries Journal.*

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of Contents

1. Introduction 1

1.1. Motivation 1

1.2. Research Aim 2

1.3. British Telecom 2

1.4. Business Processes 3

1.4.1. Types of Business Process 3

1.4.2. Business Process Reengineering in the Service Sector 5

1.5. Research Questions 6

1.6. Introduction to the Key Research Areas 7

1.6.1. Process Representation 8

1.6.2. Process Data 9

1.6.3. Automated Business Process Reengineering 9

1.7. Structure of Thesis 9

2. A Review of Literature 13

2.1. Business Process Modelling 14

2.1.1. Business Process Modelling Techniques 16

2.2. Business Process Reengineering 27

2.2.1. Application of BPR 32

2.3. Business Process Optimisation 37

2.3.1. Single and Multi-Objective Optimisation 39

2.3.2. Optimisation Algorithms 39

2.4. Cost Estimation/Modelling 45

2.4.1. Cost Estimating 45

2.4.2. Cost Estimating Techniques 46

2.4.3. Cost Estimating Techniques Summary 50

2.5. Customer Satisfaction 51

2.6. Classification Techniques 53

2.6.1. Classification and Coding Systems 53

2.6.2. Group Technology 54

2.6.3. Taxonomies 55

2.7. Relational Database Design 56

2.8. Summary of Research Findings 57

2.9. Summary and Discussion 58

2.9.1. Process Modelling 58
2.9.2. Process Analysis, Optimisation and Reengineering 60
2.9.3. Process Evaluation 62
2.10. Summary of Research Gaps 63
3. Research Aim, Objectives and Methodology 65
3.1. Research Aim and Objectives 65
3.1.1. Aim 65
3.1.2. Objectives 65
3.2. Development of a Research Methodology 66
3.2.1. Types of Research 66
3.2.2. Selected Approach 68
3.2.3. Data Collection 69
3.2.4. Research Evaluation 72
3.3. Overall Methodology Adopted 73
3.4. Summary 76
4. Methodology for Business Process Representation 77
4.1. Current Research and Development at British Telecom 77
4.2. British Telecom – Case Studies & Workshop 79
4.3. Business Process Representation 80
4.4. Business Process Representation Methodology 82
4.4.1. Taxonomy 82
4.4.2. Taxonomy Coding Mechanism 84
4.4.3. Process Coding Mechanism 85
4.5. Process Library 90
4.6. Translation of Process Models from ARIS and IDEF3 94
4.7. Summary 96
5. Automated Business Process Reengineering Methodology 97
5.1. Methodology 98
5.1.1. Initial Stage 100
5.1.2. Reengineering Stages 101
5.1.3. Optimisation Stage 114
5.1.4. Final Stage 123
5.2. Summary 123
6. Probability Driven Activity Based Cost Estimation of Business Processes 125
6.1. Industry Survey 126
6.2. ABC Forum 129

6.3. ABC Methodology Developed ____ 130
6.3.1. Sequence Flow Loops ____ 135
6.3.2. Forks and Joins ____ 139
6.3.3. ABC Module ____ 141
6.4. Validation of ABC Methodology ____ 145
6.5. Summary ____ 146
7. Methodology for Measuring Impact on Customer Satisfaction ____ 147
7.1. Interviews ____ 148
7.2. Customer Satisfaction Impact Methodology ____ 149
7.2.1. Example One – Cycle Time ____ 151
7.2.2. Example Two – Right First Time ____ 153
7.2.3. Example Three – The Two Times ____ 154
7.3. Validation of Customer Satisfaction Methodology ____ 156
7.4. Summary ____ 157
8. Integration and Implementation of Automated Business Process Reengineering Approach ____ 159
8.1. Integration via Prototype Software Platform ____ 160
8.1.1. Users ____ 160
8.1.2. User Interface ____ 160
8.1.3. Inputs ____ 161
8.1.4. Outputs ____ 161
8.1.5. Database Development ____ 161
8.1.6. Functionality ____ 164
8.2. Tour of Prototype Software Platform: ABPRS ____ 168
8.2.1. Main Menu ____ 169
8.2.2. Reengineering Programs ____ 169
8.2.3. Taxonomy Details ____ 172
8.2.4. Process Details ____ 173
8.2.5. Cost Data ____ 175
8.3. Summary ____ 176
9. Results and Validation ____ 179
9.1. Test Scenarios ____ 179
9.1.1. Test Scenario 1: Modify Order Process ____ 182
9.1.2. Test Scenario 2: Credit Application Process ____ 185
9.1.3. Test Scenario 3: Automated/Online Order Process ____ 189
9.1.4. Test Scenario 4: Telephone Order Process ____ 193

9.1.5. Test Scenario 5: Generic Processes 198
9.2. Expert Opinion 203
9.2.1. Interviewee Profiles 204
9.2.2. Process Coding Mechanism 204
9.2.3. Automated Reengineering 205
9.2.4. Cost Estimation 206
9.2.5. Customer Satisfaction 207
9.2.6. Data 208
9.2.7. Integrated Technique 209
9.3. Analysis of Results 210
9.3.1. Process Coding Mechanism 211
9.3.2. Automated Reengineering 211
9.3.3. Cost Estimation 212
9.3.4. Customer Satisfaction 213
9.3.5. Data 213
9.3.6. Integrated Technique 214
9.4. Summary 215
10. Discussion and Conclusions 217
10.1. Summary of the Achievement of Aim and Objectives 217
10.1.1. Achievement of Objective 1 217
10.1.2. Achievement of Objective 2 219
10.1.3. Achievement of Objective 3 219
10.1.4. Achievement of Objective 4 220
10.1.5. Achievement of Objective 5 220
10.1.6. Achievement of Objective 6 221
10.1.7. Achievement of Research Aim 221
10.2. Research Contributions 221
10.3. Limitations 223
10.3.1. Process Coding Structure and Methodology 223
10.3.2. Automated Reengineering Methodology 224
10.3.3. ABC Structure and Methodology 224
10.3.4. Customer Satisfaction Methodology 225
10.3.5. Integrated Approach - ABPRS 225
10.4. Business Impact Analysis 226
10.4.1. Business Process Data 226
10.4.2. Maintenance 227

10.4.3. Required Resources 227
10.4.4. System Usage 228
10.5. Further Work 229
10.6. Conclusion 230
References 235
Appendix A: Cost Estimating Survey 249
Appendix B: Customer Satisfaction Questionnaire 257
Appendix C: Validation Questionnaire 261
Appendix D: Database Tables 275

THIS PAGE IS INTENTIONALLY LEFT BLANK

List of Figures

Figure 1: Research Areas..8
Figure 2: Thesis Structure 11
Figure 3: Technique Components in Relation to Literature Reviewed....................14
Figure 4: Simple Flow Diagram 17
Figure 5: RAD describing the behaviour of a barber (Walters, 2005) 18
Figure 6: IDEF3 Process Model (Wainwright, 2004)...19
Figure 7: ARIS Process Model (as supplied by industrial sponsor) 22
Figure 8: Business Process Model in JBoss jBPM..23
Figure 9: Example BPR Methodology (Towers, 1994)..33
Figure 10: Four Levels of BPR (Burke and Peppard, 1994) 34
Figure 11: Popularity Comparison of Optimisation Algorithms (Engineering Village, 2008)...45
Figure 12: ABC Steps...48
Figure 13: Example Customer Satisfaction Survey Structure (Mihelis et al, 2001) 52
Figure 14: Suggested Classification and Coding Hierarchy......................................54
Figure 15: Possible Business Process Taxonomy..55
Figure 16: Entity Relationship Diagram 56
Figure 17: Research Programme 74
Figure 18: Current R & D at British Telecom – JBoss jBPM...................................78
Figure 19: Example Service Sector Taxonomy..83
Figure 20: Activity ‘Modify Order’ 85
Figure 21: Modify Order Process Model...85
Figure 22: Modify Order - Task Code...90
Figure 23: Process Library...93
Figure 24: ARIS Translation to BPMN Flow Diagram 95
Figure 25: IDEF3 Translation to BPMN Flow Diagram 95
Figure 26: Automated Reengineering Methodology - Program Structure Chart ..99
Figure 27: Remove Duplicates/Redundant Steps – Process Before 102
Figure 28: Remove Duplicates/Redundant Steps – Process After 103
Figure 29: Section Replaced with Single Step – Process Before............................106
Figure 30: Section Replaced with Single Step – Process after Forward Search...108
Figure 31: Section Replaced with Single Step – Process after Backward Search 108
Figure 32: Replacement Sub-Process 111
Figure 33: Section Replaced with Sub-Process – Before...111
Figure 34: Section Replaced with Sub-Process – After...113
Figure 35: Optimisation Stage Flowchart 115
Figure 36: Additional/Overhead Cost Allocation (as supplied by industrial sponsor)...131
Figure 37: Cost Data...132
Figure 38: Credit Check Process...133
Figure 39: Credit Check Process – Costs & Probabilities 134
Figure 40: Sequence Flow Loop – Type 1..136
Figure 41: Sequence Flow Loop – Type 2..137
Figure 42: Sequence Flow Loop – Type 2 Example 138

Figure 43: Sequence Flow Loop – Type 3....139
Figure 44: Fork and Join Gateways....140
Figure 45: Telephone Order Process 144
Figure 46: Example 1 – Existing Serial Process....151
Figure 47: Example 1 – First Alternative Process....151
Figure 48: Example 1 – Second Alternative Process....152
Figure 49: Example 2 – Existing Process with Gateway....153
Figure 50: Example 2 – First Alternative Process....153
Figure 51: Example 2 – Second Alternative Process....154
Figure 52: Example 3 – Existing Process with Gateway....154
Figure 53: Example 3 – First Alternative Process....155
Figure 54: Example 3 – Second Alternative Process....156
Figure 55: Components of Automated Reengineering Approach....159
Figure 56: Database Schema Diagram 163
Figure 57: Example Process Data 164
Figure 58: Modular Structure of ABPRS....165
Figure 59: Objective Value Relationships....167
Figure 60: ABPRS Architecture....169
Figure 61: ABPRS - Main Menu....169
Figure 62: ABPRS - Reengineering Programs Menu....170
Figure 63: ABPRS - Reengineering Algorithm....170
Figure 64: ABPRS – Reengineered Process Report 171
Figure 65: ABPRS – Reengineered Process Chart....171
Figure 66: ABPRS - Taxonomy Details Menu....172
Figure 67: ABPRS - Taxonomy Details....172
Figure 68: ABPRS - Company Taxonomy Diagram....173
Figure 69: ABPRS - Process Details Menu 173
Figure 70: ABPRS - Process Model Selection....174
Figure 71: ABPRS - Process Model Form/Report....174
Figure 72: ABPRS - Available Process Step Form....175
Figure 73: ABPRS - Cost Data Menu....175
Figure 74: ABPRS - Activity Centre Costs Form....176
Figure 75: ABPRS - Resource Centre Costs Form....176
Figure 76: Software Platform – GA Settings 180
Figure 77: Test Scenario 1 – Existing Modify Order Process 182
Figure 78: Test Scenario 1 –Modify Order Process Before....183
Figure 79: Test Scenario 1 –Modify Order Process After 185
Figure 80: Test Scenario 1 – Changing Cost per Process Input to Optimisation Stage....185
Figure 81: Test Scenario 2 – Existing Credit Application Process 186
Figure 82: Test Scenario 2 –Credit Application Process Before....187
Figure 83: Test Scenario 2 –Credit Application Process After 188
Figure 84: Test Scenario 2 – Changing Cost per Process Input to Optimisation Stage....189
Figure 85: Test Scenario 3 – Existing Automated/Online Order Process....189
Figure 86: Test Scenario 3 –Automated/Online Order Process Before....190
Figure 87: Test Scenario 3 –Automated/Online Order Process After....192

Figure 88: Test Scenario 3 – Changing Cost per Process Input to Optimisation Stage..........193
Figure 89: Test Scenario 4 – Existing Telephone Order Process..........194
Figure 90: Test Scenario 4 –Telephone Order Process Before..........195
Figure 91: Test Scenario 4 –Telephone Order Process After..........198
Figure 92: Test Scenario 4 – Changing Cost per Process Input to Optimisation Stage..........198
Figure 93: Test Scenario 5 – Generic Process..........199
Figure 94: Test Scenario 5 –Generic Process Before..........201
Figure 95: Test Scenario 5 –Generic Process After..........201
Figure 96: Test Scenario 5 –Generic Process Sub Scenario Comparison 1 to 4...202
Figure 97: Test Scenario 5 –Generic Process Sub Scenario Comparison 1,5 and 6 203
Figure 98: Final Contributions Summary..........223

THIS PAGE IS INTENTIONALLY LEFT BLANK

List of Tables

Table 1: Business Process Modelling Techniques (Aguilar-Saven, 2004) 25
Table 2: Guidelines for Selecting Tools/Techniques for BPR (Gunasekaran and Kobu, 2002) 30
Table 3: Cost Estimating Techniques Summary 50
Table 4: Quantitative Approach: Strengths and Weaknesses...................................67
Table 5: Qualitative Approach: Strengths and Weaknesses 67
Table 6: Comparison of Qualitative and Qualitative Research Strategies (Burns, 2000)...68
Table 7: Strengths and Weaknesses of Interviewing...70
Table 8: Disadvantages and Advantages of Surveys (Robson, 2002)........................71
Table 9: Taxonomy Coding System 84
Table 10: Process Code Details...86
Table 11: BPMN Flow and Step Types...88
Table 12: Industry Survey - Relevant Points 129
Table 13: Example Tasks and Resources 131
Table 14: Credit Check Process – Cost Estimate 134
Table 15: Fork and Join Process – Cost Estimate 140
Table 16: Telephone Order Process – Cost Estimate..145
Table 17: Entity Relationship Summary..162
Table 18: Validation – Average Respondent Profile 204

THIS PAGE IS INTENTIONALLY LEFT BLANK

List of Abbreviations

ABC	Activity Based Cost Estimation
ABPRS	Automated Business Process Reengineering System
ARIS	Architecture of Integrated Information Systems
BPML	Business Process Modelling Language
BPMN	Business Process Modelling Notation
BPMS	Business Process Management System
BPO	Business Process Optimisation
BPR	Business Process Reengineering
BT	British Telecom
CPN	Coloured Petri Net
CRM	Customer Relationship Management
DFD	Data Flow Diagram
EA	Evolutionary Algorithm
EC	Evolutionary Computing
EDF	Electricite de France
EPSRC	Engineering and Physical Sciences Research Council
ER	Entity Relationship
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
GP	Genetic Programming
GT	Group Technology
GUI	Graphical User Interface
IDEF	Integrated Definition for Function Modelling
JPDL	JBoss Process Definition Language
MOO	Multi Objective Optimisation
MS	Microsoft
NSGA	Non-Dominated Sorting Genetic Algorithm
OMG	Object Management Group
RAD	Role Activity Diagram
RDB	Relational Database
RID	Role Interaction Diagram
SOA	Service Oriented Architecture
SOO	Single Objective Optimisation
SPEA	Strength Pareto Evolutionary Algorithm
SQL	Structured Query Language
UML	Unified Modelling Language
UPS	United Parcel Service
VB	Visual Basic
WFMC	Workflow Management Coalition
WFMS	Workflow Management System

THIS PAGE IS INTENTIONALLY LEFT BLANK

1. Introduction

This chapter sets the scene for the research project. It begins by describing the motivation for the project and its overall aim, and introduces and provides an overview of the industrial sponsor, British Telecom. Business processes, what they are and the different types are introduced, along with the reasons for reengineering them. The research questions derived from the motivation and aim are presented, and the research areas investigated in order to achieve the aim, and based on the research questions, are also summarised. Finally, the structure of the thesis is described with a brief description of each chapter.

1.1. Motivation

This research project involves the development of an automated business process reengineering (BPR) methodology for the service sector. It is sponsored by the Engineering and Physical Science Research Council (EPSRC) and British Telecom (BT). EPSRC is the UK government's leading funding agency for research and training and invests approximately £740 million per annum in subject areas such as engineering, mathematics, materials science and information technology. BT is the principal telecommunications service provider in the UK and the main beneficiary of this project. An automated business process reengineering framework will enable BT to improve their customer service by reducing the cost of their services and potentially increase, or at least maintain, their levels of customer satisfaction, and provide them with an 'intelligent' decision support for making decisions about their process designs. The new knowledge developed through the research will also help other companies in the service sector.

Other service sector companies were also involved in this research in order to derive a comprehensive overview (HSBC and Clydesdale Banks, Electricite de France (EDF)/Sainsburys Energy, plus four separate BT business units), and the resultant framework was then validated principally with British Telecom along with

EDF/Sainsburys Energy, ACIS, United Parcel Services (UPS) and Cambridgeshire Constabulary (Citizen Experience Project).

1.2. Research Aim

The research aim is to develop a framework to automate business process reengineering in the service sector using process cost, duration and impact on customer satisfaction as criteria.

1.3. British Telecom

BT is the world's oldest communications company, and is directly descended from the first commercial telecommunications undertaken in Europe. The Electric Telegraph Company, established in 1846, was the first organization outside the USA to exploit leading edge telegraphy technology and introduce electrical communications to the world. Within ten years, an international network had been developed, making communications possible within minutes instead of days and weeks. The consequences for every aspect of society were dramatic and profound.

The BT brand became established in 1980 as the official name of Post Office Telecommunications, becoming a state owned organisation independent of the post office. At this time, BT was the UK’s only telecommunications provider. This changed in 1982 when their monopoly was broken by Mercury Telecommunications who were also granted a licence to provide telecommunications services. Presently, there are multiple telecommunications services in operation throughout the UK. As a result, technology is more central to BT's business than ever before - virtual markets, electronic commerce, broadband and mobility are now radically changing the way companies and people conduct business. The successful companies of the future will be those which exploit technology to underpin their business, generate revenues and minimise costs. For BT, the willingness to embrace new relationships, both technical and commercial, will be key to maintaining its influence on the development of the communication industry (Duxbury et al, 1999).

BT is one of the largest telecommunications service producers in the world with over 38,500 employees worldwide, and has service operations in many countries. Its principal activities include networked IT services, local, national and international telecommunications services, and higher value broadband and internet products and services. In the UK, BT serves more than 20 million business and residential customers with more than 30 million exchange lines, as well as providing network services to other licensed operators. The group is formed of different businesses which include BT Retail, BT Global Services, BT Wholesale and Openreach.

1.4. Business Processes

As business processes and their automated reengineering are the basis of this research, it is necessary to introduce them and provide a definition. There are many definitions of a business process: it can be described as “A set of logically related tasks performed to achieve a defined business outcome” (Davenport and Short, 1990) or, alternatively, “A set of activities, which can be broken down into tasks, that when taken together take an input, transform it, and produce an output” (Johansson et al, 1993), or “a specific group of activities and subordinate tasks that results in the performance of a service that is of value” (Johnson and Weinstein, 2004). In order to reengineer business processes, it is necessary to examine some of their relevant aspects with particular regard to their representation – crucial because the process representation must be quantitatively analysed before it can be reengineered. It is then necessary to investigate how to store, retrieve and manipulate an existing business process.

1.4.1. Types of Business Process

All organisations operate via the use of business processes. These processes may not be formalised and represented but they will always exist, regardless of the type of business which the organisation is involved in (Stelling et al, 2008a).

There are arguably three broad types of business process: production (for example, taking raw material and transforming it into a finished product), office (more ‘goal based' where people do whatever is necessary to attain a goal) and management (decision making).

Melao and Pidd use metaphors to specifically describe business processes and give four perspectives on business process to develop a conceptual framework with which to understand business processes more fully. The four categories are:

1. ***Business processes as deterministic machines***. This view of business processes emphasises the structure (tasks, activities, areas of responsibility), procedures (constraints and rules regarding the work to be performed) and goals (type of output) of the business process being designed. The main criterion is efficiency in the use of money, resources and time, subject to the constraint of satisfying customers' requirements. I.T. plays an important role in this type of process. This view relates to Davenport and Short's definition (above) of a business process as "A set of logically related tasks performed to achieve a defined business outcome".

2. ***Business processes as complex dynamic systems***. Unlike the mechanistic view above, this view pays much more attention to interactions with the external environment - quality and service level are major process design criteria rather than just efficiency.

3. ***Business processes as interacting feedback loops***. This view claims that business processes are closed loops with intrinsic control – it is an attempt to understand the dynamic behaviour of a process in terms of interactions between internal structure and policy rather than in terms of individual components (i.e. a network of interacting feedback loops)

4. ***Business processes as social constructs***. This view emphasises business processes as made and enacted by people with different values, expectations and agendas – the focus is on subjective and human aspects of the process. It lends itself to less tangible processes where human activity is the main driver (such as health and education services)

These four categories of business process are not independent of each other - there is a degree of over-lap between them and so a good process design should take into consideration more than one of them (Melao and Pidd, 2000).

Business processes can also be categorised into serial or parallel and simple or complex. Serial processes progress in a straight line with one step following another, whereas parallel processes may branch off and involve a number of steps performed in situ. Simple or complex is fairly self explanatory: a simple process may be a serial process consisting of only a few steps whereas complex processes may involve an infinite number, perhaps branching off on alternative routes depending on numerous criteria with many parallel routes being performed at the same time.

1.4.2. Business Process Reengineering in the Service Sector

There has been a varying pattern in the perception of the 'best' process over the last 40 years: In the 1960's industry concentrated on how to produce more (quantity); in the 1970's how to produce it cheaper (cost); in the 1980's how to produce it better (quality); in the 1990's how to produce it quicker (lead time); and in the 21st century how to offer more (service) (Lindsay et al, 2003). There has always been a desire to make processes as efficient as possible (Stelling et al, 2006) - ideally, a process should possess all of the above qualities, with minimum cost and duration, and maximum customer value (Stelling et al, 2009).

Driven by the ideal qualities mentioned above, business processes are reengineered – to save time and money, to retain existing customers due to the quality of the product or service provided, and to attract a multitude of new customers for these same reasons. There are many definitions of business process reengineering (BPR), some of which are presented in Chapter 2. A well-known and suitable one for this research defines BPR as "the fundamental rethinking and redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance such as cost, quality, service and speed" (Hammer and Champy, 1993).

Most processes are designed to be relatively simple, but they can grow complex over time as modifications and additions are made to deal with exceptions or special cases, leading to complicated and inefficient processes (Hammer and Champy, 1993) which require improvements. BPR is a well known procedure throughout industry, and is generally performed by human experts, invariably assisted by various software tools, capable of making judgements concerning potential process improvements. BPR is reviewed in Chapter 2.

The research involved in this project is aimed particularly at service based business (or office) processes, although it is also relevant to some other types of processes. Certain business processes, specifically service based business processes, differ to those in other industries, principally because they contain human interaction (i.e. human to human via the telephone or face to face, human to machine via web-based processes) and involve a customer – decisions may be made or events may occur during their execution which may influence the path that they take, and perhaps involve repetitions of sections of processes, resulting in processes which may be variable in nature (Burke and Peppard, 1994), (Razavian and Ramtin, 2008) and which sometimes follow numerous, different routes. This fact leads to a different method of evaluation with regard to process cost, duration and impact on customer satisfaction. Examples of service sector processes include fault reporting, requesting advice or assistance, placing an order, billing queries and service upgrades.

1.5. Research Questions

In order to form a basis for the research, four key research questions were formulated during the early stages of the project, each of which has guided the project through its various levels and resulted in the development of the automated BPR technique. They are as follows:

- In order to reengineer a business process via the use of conventional and/or evolutionary computation, a process model must be represented in a way that allows its input to, analysis and evaluation by an algorithm. How can this be done comprehensively and in a way which caters for complex process models?

- Is it possible to use existing optimisation algorithms for this type of reengineering and optimisation?

- Business processes may follow a number of different routes based on the outcome of events or decisions. Therefore, how can the cost of a business process be predicted and how can this prediction be automated?

- Customer satisfaction levels are generally measured using feedback from customers. Without the facility to collect or analyse feedback, how can this measurement be automated?

Therefore, these four questions formed the basis of the overall research project and, by attempting to answer all of them, a framework for the automated reengineering methodology was created. The key research areas investigated in order to answer the research questions are introduced in Section 1.6.

1.6. Introduction to the Key Research Areas

Guided by the overall aim and research questions of the project, it was necessary to conduct research into a relatively wide ranging number of subject areas, all of which are reviewed and discussed in Chapter 2. The following diagram, Figure 1, illustrates the areas covered and how they link to the three main elements of the overall reengineering technique: process representation, process data and the actual technique.

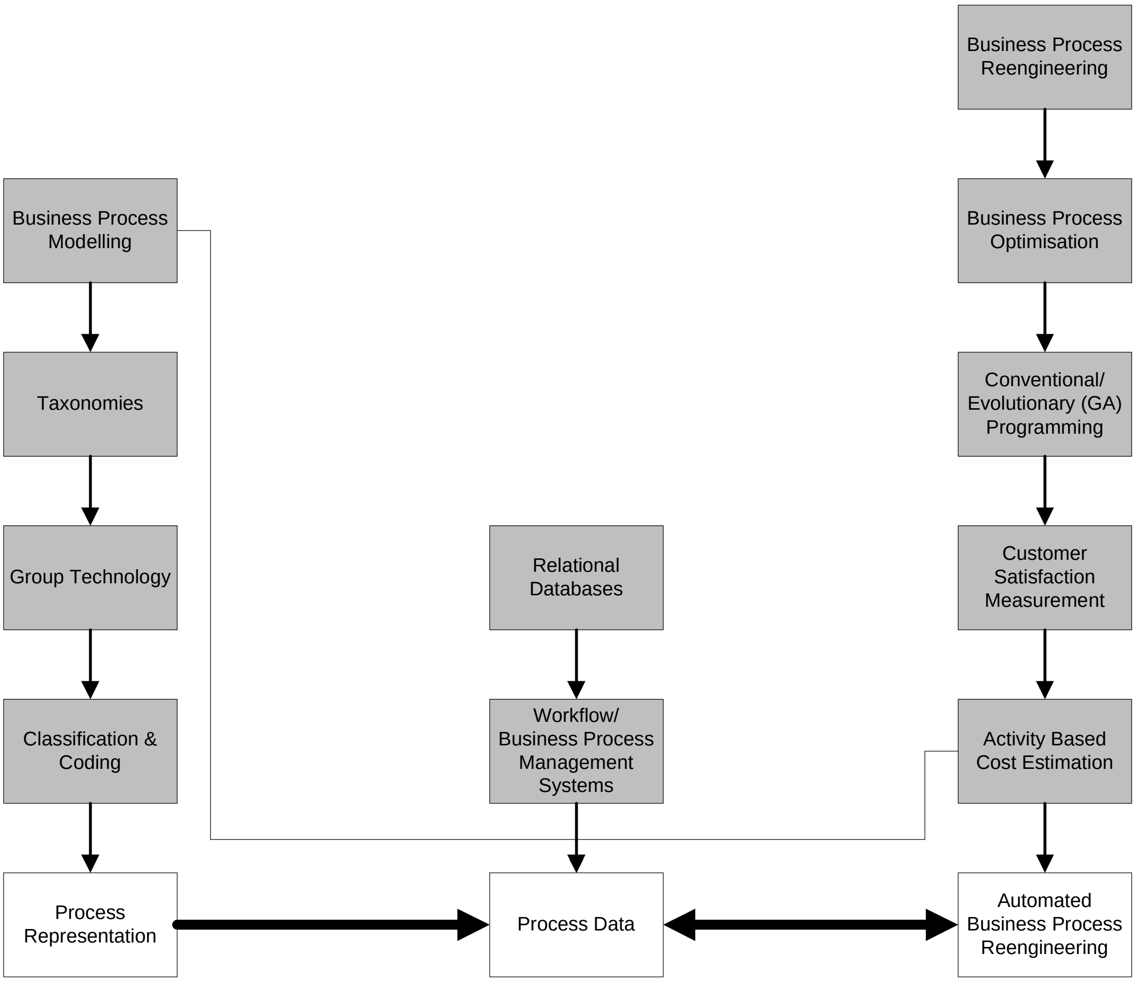

Business Process Reengineering
Business Process Modelling
Business Process Optimisation
Taxonomies
Conventional/ Evolutionary (GA) Programming
Group Technology
Relational Databases
Customer Satisfaction Measurement
Classification & Coding
Workflow/ Business Process Management Systems
Activity Based Cost Estimation
Process Representation
Process Data
Automated Business Process Reengineering

Figure 1: Research Areas

1.6.1. Process Representation

The process representation is the input to the automated reengineering technique, and it should be possible to store and retrieve all process representations. They should also contain sufficient information for the technique to reengineer/optimise and analyse/evaluate them. As such, a review of process modelling techniques was an initial requirement. This then led to other areas which became of relevance due to the findings of the modelling review – notably taxonomies, group technology, classification and coding systems, and workflow and business process management systems (WFMS and BPMS).

1.6.2. Process Data

As mentioned in the previous section, it should be possible to store and retrieve all process representations. They should also contain sufficient information for analysis and evaluation based on cost, duration and impact on customer satisfaction. A relational database was the foundation for this, with ideas provided by WFMS and BPMS. It was also necessary to allow for a cost estimating function within the database structure.

1.6.3. Automated Business Process Reengineering

Following on from the previous two elements of the overall technique, the areas of BPR and business process optimisation (BPO) were reviewed in order to gain ideas as to how the technique should function. The main levels and guidelines of BPR were identified, followed by an investigation into Evolutionary Computing (EC) which provided an insight into the benefits of Genetic Algorithms (GA) when applied to search and optimisation problems. This, coupled with research into cost estimating and customer satisfaction measurement, led to ideas concerning the overall functionality of an automated reengineering technique, potentially extended for purposes of increased effectiveness to include an evolutionary based optimisation module.

1.7. Structure of Thesis

The thesis is structured as follows:

Chapter 1 provides a summary of the research project, including the motivation for and aim of the project; an overview of the industrial sponsor, British Telecom; an introduction to business processes, the different types, why service sector processes are different, and reasons for reengineering; the research questions of the project; and it briefly lists the areas of research and how they fit in with the main areas of the project.

Chapter 2 is a review of the relevant literature which all contributes in varying degrees to the development of the automated BPR technique and its component parts. Each subject area is explained and discussed concerning its context within the project. The research findings and gaps, summarised in Section 2.8 and 2.10 respectively, are also highlighted.

Chapter 3 concerns the development of the research aim, objectives and methodology. Principles of research design are reviewed and the research methodology adopted for this project is detailed. How the next four chapters, 4-7, fit in is also explained.

Chapter 4 Provides a background to the methodologies developed in this research, and described in this chapter and the following three chapters, including a description of current research and development work by the industrial sponsors (case studies), and details of a workshop conducted with them. It then goes on to explain the development of the business process representation methodology – the proposed framework and mechanism for creating numeric process models is described, some of which is based on the case studies.

Chapter 5 explains the development of the automated reengineering framework and provides details of the development of the hybrid algorithm.

Chapter 6 explains the development of the activity based costing (ABC) structure and methodology, and how it incorporates with the reengineering algorithm, along with details of an industry survey conducted prior to development which assisted in the creation of the technique, and an online forum which verified the requirement for the technique.

Chapter 7 explains the development of the customer satisfaction change detection methodology and how standard techniques could not be used. The criteria and algorithm created to achieve this are described, which includes the results from interviews with experts. How the methodology is incorporated with the reengineering and optimisation algorithm is also included.

Chapter 8 explains the complete automated reengineering approach and the integration with all its component methodologies. The database and the prototype software platform are presented here in detail.

Chapter 9 provides detailed process reengineering test scenarios, both generic and real-life, in order to demonstrate quantitative validation of the research. Qualitative validation is then demonstrated via the results of a questionnaire on the research and development, completed by a number of experts.

Chapter 10 is a discussion of the research project, including contributions of the research, limitations of the research, impact on businesses, and suggestions for further improvements and work.

The thesis structure with links between chapters and their sources of data, where applicable, is illustrated in Figure 2.

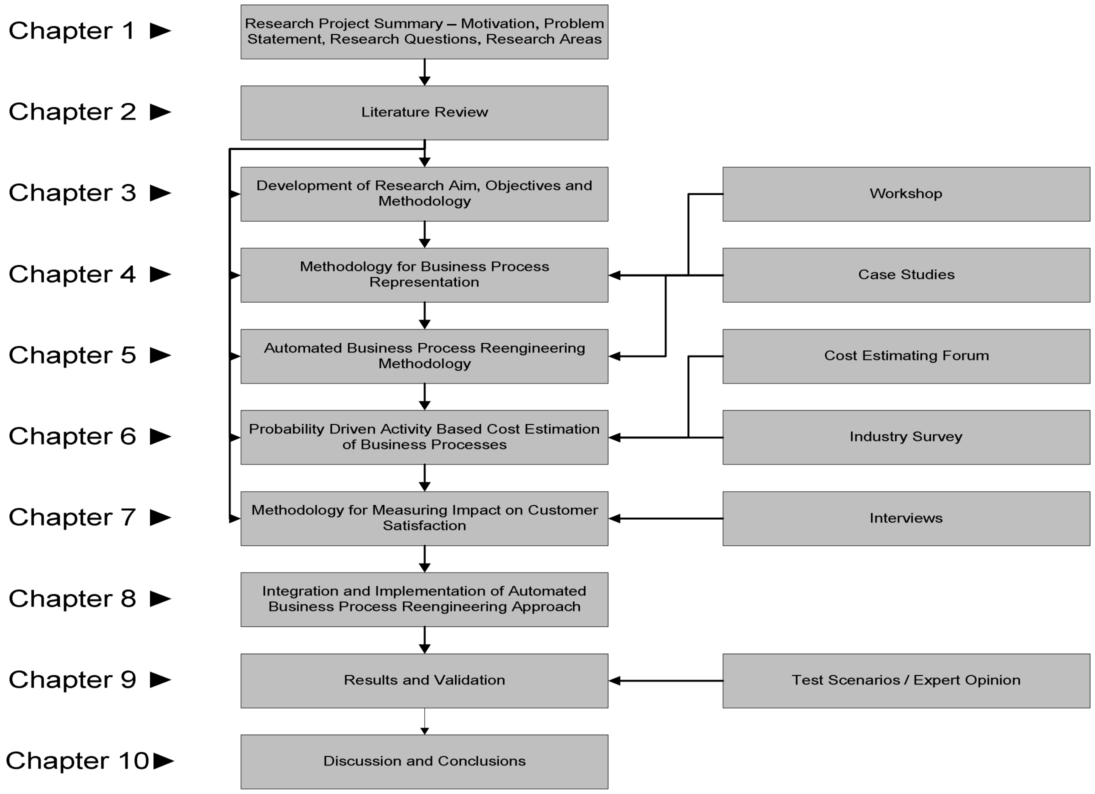

Chapter 1 ►
Research Project Summary – Motivation, Problem Statement, Research Questions, Research Areas
Chapter 2 ►
Literature Review
Chapter 3 ►
Development of Research Aim, Objectives and Methodology
Workshop
Chapter 4 ►
Methodology for Business Process Representation
Case Studies
Chapter 5 ►
Automated Business Process Reengineering Methodology
Cost Estimating Forum
Chapter 6 ►
Probability Driven Activity Based Cost Estimation of Business Processes
Industry Survey
Chapter 7 ►
Methodology for Measuring Impact on Customer Satisfaction
Interviews
Chapter 8 ►
Integration and Implementation of Automated Business Process Reengineering Approach
Chapter 9 ►
Results and Validation
Test Scenarios / Expert Opinion
Chapter 10►
Discussion and Conclusions

Figure 2: Thesis Structure

THIS PAGE IS INTENTIONALLY LEFT BLANK

2. A Review of Literature

This part of the thesis forms the background to the project. A range of areas have been reviewed during the course of the project in order to gain understanding and ideas, and to identify research gaps and shortfalls. Section 1.6 briefly discussed these areas, and Figure 1 illustrated how they link. The literature review was conducted on an ongoing basis during the project, but contained two major phases with a period of experimentation in between (see Chapter 3 for details). The two phases were as follows:

Phase 1	Business Process Modelling
	Business Process Reengineering
	Business Process Optimisation
	Evolutionary Computing
	Cost Estimation
	Workflow and Business Process Management Systems
Phase 2	Taxonomies
	Group Technology
	Classification and Coding
	Relational Databases
	Customer Satisfaction Measurement

The technique developed comprises a number of separate components, some of which could be used independently of the others. Figure 3 attempts to illustrate the flow and structure of the technique, and the links between the areas reviewed and the components of the technique.

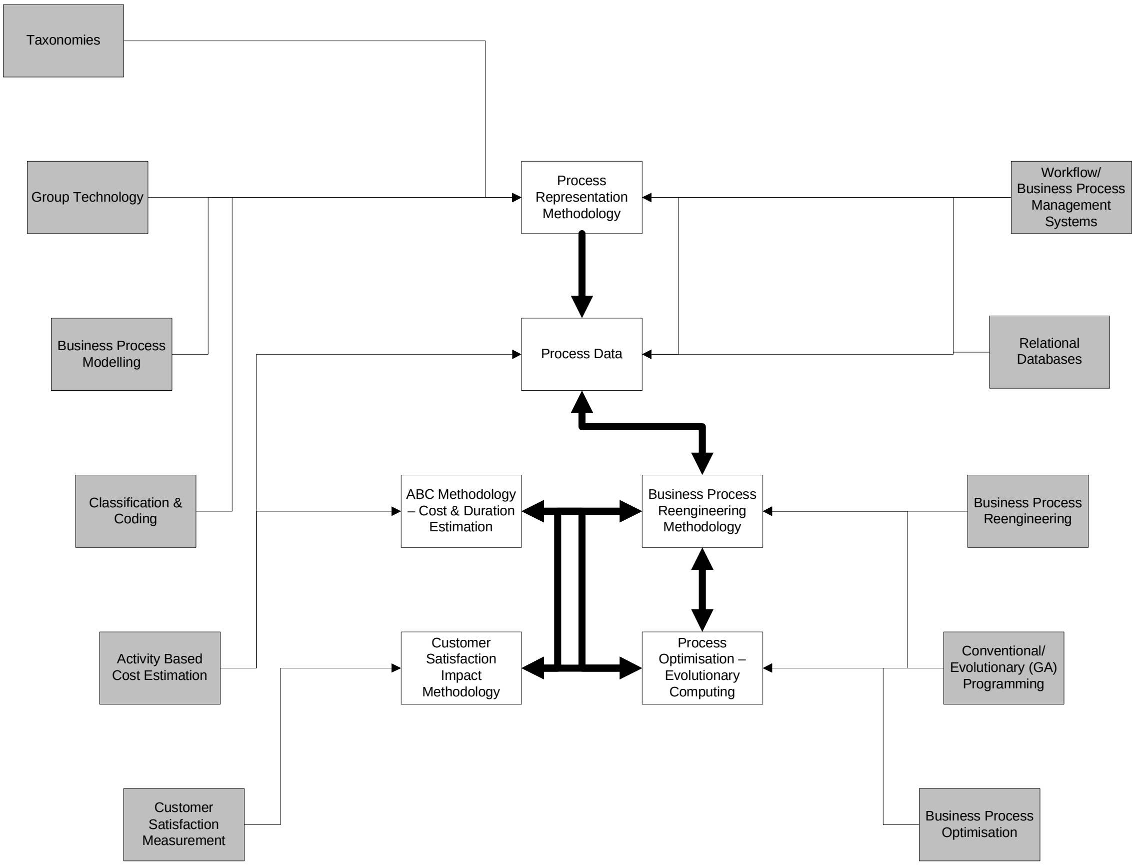

Taxonomies
Group Technology
Process Representation Methodology
Workflow/ Business Process Management Systems
Business Process Modelling
Process Data
Relational Databases
Classification & Coding
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Business Process Reengineering
Activity Based Cost Estimation
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing
Conventional/ Evolutionary (GA) Programming
Customer Satisfaction Measurement
Business Process Optimisation

Figure 3: Technique Components in Relation to Literature Reviewed

Each of the above areas is reviewed in varying detail with an indication of their relevance and contribution to the project, followed by a final summary and overall discussion.

2.1. Business Process Modelling

Representing a business process by means of modelling enables a common understanding and analysis of the process and can provide a comprehensive familiarity with the process. Also, an enterprise can be analysed and integrated through its business processes. It is therefore crucial to correctly model business processes and to take into account the uses or purposes of the model in order to select the most appropriate technique (Aguilar-Saven, 2004). Process modelling is a complicated and error-prone procedure, and it is generally accepted in both research and industry that the cost of modifying errors is very high after a process definition becomes operational (Jiantao et al, 2005). Certainly, in service based processes, there is a need to explore in detail the ‘flow’ between a customer request and eventual satisfaction with the goods/services he’s provided with - within that flow, there is the need to understand where defects and/or unnecessary delays occur so the process can be adjusted in order to remove them (Ould, 1995).

Process modelling tools allow for a systematic representation of processes in manufacturing, product development, and service applications. The simplest and perhaps most illustrative way of representing a process is by drawing a flow diagram, but there are numerous techniques and tools available: some of the more widely used ones in industry are Integrated Definition for Function Modelling (IDEF), Unified Modelling Language (UML), Petri Nets (Zakarian and Kusiak, 2000) (Gordijn et al, 2000) and Architecture of Integrated Information Systems (ARIS). In fact, Lindsay et al (Lindsay et al, 2003) argue that that the lack of standardisation in representing a business process is partly caused by the absence of an adequate definition of a business process. However, modelling standard Business Process Modelling Notation (BPMN), developed by the Object Management Group (OMG), aims to provide a notation that is readily understandable by all business users, and its intent is to standardise business process modelling notation (Object Management Group (OMG), 2006).

As a suitable process representation is absolutely crucial for the development of the reengineering and optimisation approach, and given that there are a very large number of different modelling tools and techniques available, initial process modelling criteria for the purposes of this project (and certainly more widely) were determined as follows in order to target the review more effectively:

- The process representation should be readable and understandable without specialised knowledge, such as computer science or mathematics.
- It should be possible for a user without specialised knowledge to amend or create process models (with the assistance of a graphical user interface (GUI)).
- It should allow quantitative (measurable) analysis rather than purely qualitative analysis – it is not possible to examine and optimise a process via algorithmic means

without quantitative information (for example, the duration of a task). Therefore, it should be represented with an appropriate level of detail to allow this.

- It should allow representation of complex processes - service sector business processes are based on human to human or human to machine (i.e. web-based processes) interaction and, as a result, may involve branches and/or repetitions based on decisions and events.
- The model should contain sufficient information to analyse each element of the process, individually or in sections, and compare with other process data held in a data warehouse, in order to potentially reengineer to the extent where complete process reconfiguration could occur.
- The detailed (dynamic and functional), quantitative process model should allow storage in a database and retrieval for purposes of analysis (it should be possible to store the model and then retrieve it in exactly the same state as it was stored in)
- It should be possible to link cost and duration measures to the steps in the model

2.1.1. Business Process Modelling Techniques

A selection of well-known modelling techniques, including traditional, formal/parametric and others, have been reviewed. This section attempts to provide an overview of which techniques are available and illustrate some of their characteristics, advantages and limitations, particularly regarding their suitability for use in this project.

1. ***Flow diagrams*** - a modelling technique which aims to represent a process via diagrams consisting of lines, blocks, decision boxes etc., defined as a formalised, graphic representation of a program logic sequence, work or manufacturing process, organisation chart or similar formalised structure (Lakin et al, 1996). A flow diagram can be constructed from the available symbols without the need to follow a defined methodology – this makes it an easy technique to use but can also lead to very large diagrams, causing problems in reading and navigating through the chart. Figure 4 shows a simple flow diagram:

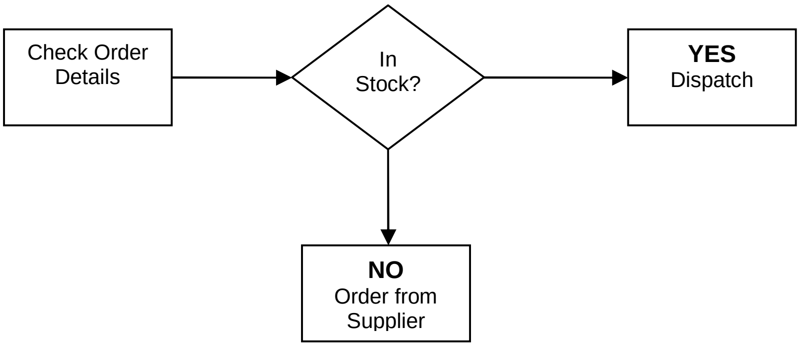


Figure 4: Simple Flow Diagram

2. ***Data Flow Technique*** **–** this technique was originally designed by Yourdon to depict data flow within a process, and how the process relates to the user and the outside world (Aguilar-Saven, 2004). As with the flow diagram technique, the data flow technique uses diagrams to represent a process. The difference with the data flow diagram (DFD) is that it depicts the flow of information within a process rather than the overall process flow – which is its main disadvantage as it does not allow for a thorough representation of the process sequence, leaving a gap in the understanding of how the process actually works.

3. ***Role Activity Diagrams (RAD)*** **-** an intuitive modelling technique with few symbols and relaxed rules, so constructing RAD models is relatively easy. RAD supports the concept of interactions between processes, which can be regarded as communications; interactions can have multiple participating processes, which can be treated as a form of broadcast communication; and they allow simultaneously executing threads and can therefore model concurrent processes (Bavan and Abeysinghe, 2005). This modelling technique, as with flow diagrams, corresponds with the view of a business process as a deterministic machine because the stress is on the mapping and documenting of the flow of items, the activities, their logical dependency and the required resources (Melao and Pidd, 2000). The diagram below, Figure 5, shows a RAD representing the process followed by a barber.

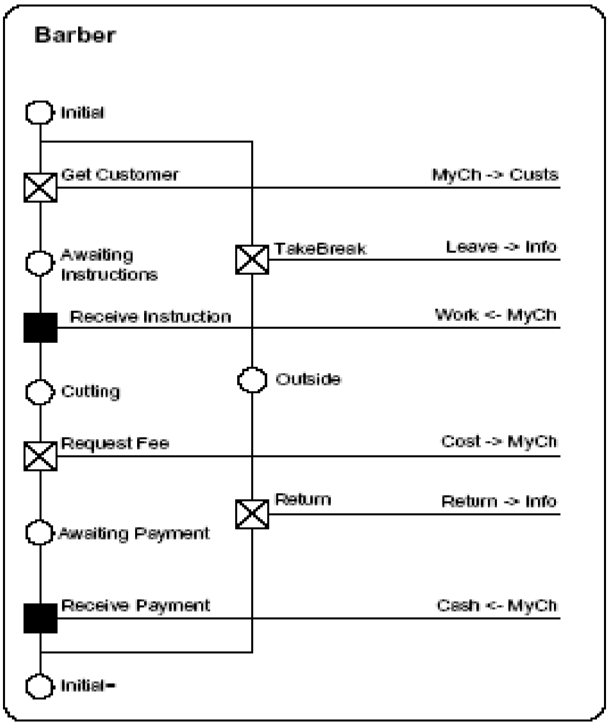


Figure 5: RAD describing the behaviour of a barber (Walters, 2005)

4. ***Role Interaction Diagrams (RID)*** - combine the characteristics of RAD and Jacobson's object interaction diagrams to produce a graphical representation of a process. Activities are connected to roles in a kind of matrix. Text and symbols are used together in order to represent the process. Although slightly more complex than flow diagrams, RID are fairly intuitive to understand and reasonably easy to read but they tend to be untidy and cluttered, making them quite hard to build. Inputs to, and outputs from the activities are not modelled which means that important information is lost - this characteristic alone makes RID unsuitable for use as a modelling technique for purposes of this project (Aguilar-Saven, 2004).

5. ***Integrated Definition for Function Modelling (IDEF)*** - originates from the Air Force program for Integrated Computer-Aided Manufacturing (ICAM) which developed the first ICAM Definition, or IDEF, methods (Mayer et al, 1995). There is a whole family of IDEF techniques each catering for different applications (IDEF0 through to IDEF14). IDEF3 was developed for explicitly describing processes – it allows the user to capture and structure descriptions of how a system works from multiple viewpoints, thereby enabling users to capture information

conveyed by knowledgeable experts about the behaviour of a system rather than directing user activity toward constructing engineering models to approximate system behaviour. The IDEF3 model is made up of units of behaviour, links, junctions, referents and notes (Bernus and Nemes, 1996). It is certainly a very useful and effective method of representing a process but, although quite rigorous, does not provide complete information. Figure 6 shows an IDEF3 process model.

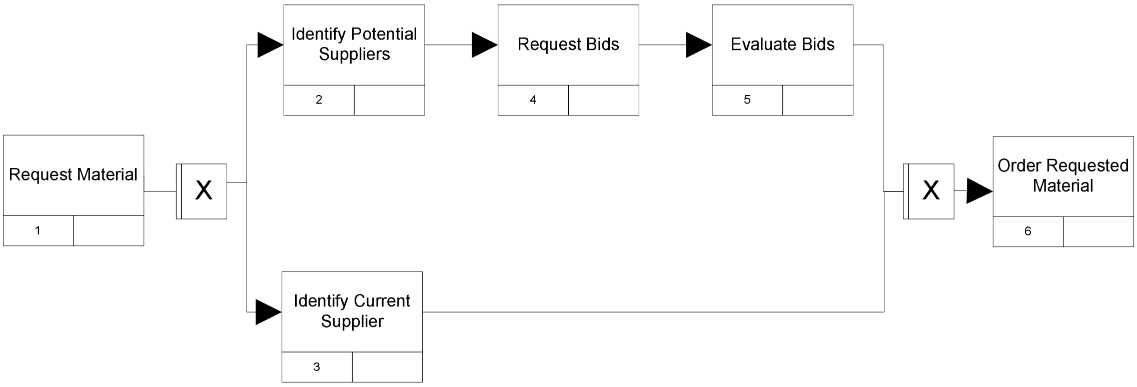


Figure 6: IDEF3 Process Model (Wainwright, 2004)

6. ***The Fuzzy Logic Technique*** - The main purpose of this technique is to remove the uncertain and incomplete information of process variables that exist in traditional techniques. This can be done by a combination of fuzzy logic and rule-based reasoning. The major benefit of this technique is the ability to quantify outputs from a process which allows for a process to be tested using different values (Zakarian, 2001). However, although it allows an element of quantitative analysis, it does not allow for the level required for a complete representation of all elements of a process.

7. ***Artificial Intelligence Formal Language*** – a formal process modelling technique which enables the production of detailed, formal specifications of business processes from high-level enterprise objectives. The use of an Artificial Intelligence formal language permits verification that the specifications possess certain correctness properties, namely that the responsibilities assigned to roles are fulfilled, and that constraints are maintained as a result of process execution. The model development is based on five sub-models that are used to formally describe different aspects of an organisation (Koubarakis and Plexousakis, 2002). Each of these models consists of

various concepts that are formally described by the declarative logical language, *L*, which was introduced by Enderton in 1972. Koubarakis and Plexousakis use *L* to introduce appropriate constructs, write axioms and capture process semantics. Their approach ensures formality of the concepts although it becomes increasingly complicated as the concepts to be modelled incorporate other concepts. The authors themselves admit that 'it is a lot of work to create and maintain a formal business process and also retain their consistency' (Vergidis, 2005). Other disadvantages of this technique are the use of complex mathematical notation, unlikely to be within the capabilities of most business analysts, along with the essential AI programming skills required.

8. ***Mathematical Definition Technique*** - uses a mathematical model with an additive objective function to represent a business process. The objective function is minimised or maximised according to an 'optimisation orientation concept' (Hofacker and Vetschera, 2001). The main concepts used in the process design are activities and resources. However, the process model needs to be further extended in several directions to justify the validity of this technique - the information available via this technique is not comprehensive enough to allow for complete, complex process representation.

9. ***Unified Modelling Language (UML)*** - an object oriented modelling technique. It represents a collection of engineering practices that have proven successful in the modelling of large and complex systems. UML covers conceptual things, such as business processes and system functions, as well as concrete things, such as programming-language classes, database schemas, and reusable software components. UML allows for the representation of a process using a common set of modelling constructs and a common notation. It consists of nine different diagrams, and each diagram shows a specific static or dynamic aspect of a system. The main advantage of UML is that it is powerful enough to specify software system models visually and efficiently. The main disadvantages are its lack of formal semantics and that it is difficult to directly apply mathematical techniques on UML

models for system validation. The information available from the UML model is also fragmented due to the number of different diagrams (Aguilar-Saven, 2004).

10. ***Coloured Petri Nets (CPN)*** - a directed graph made up of places, transitions and arcs. Places, depicted as circles, represent possible states or conditions; transitions, depicted by bars or boxes, describe events that may modify system states; and arcs represent the relationship between places and transitions (Salimifard and Wright, 2000). They can be used as a design language for the specification of complex workflows, but also Petri net theory provides for powerful analysis techniques which can be used to verify the correctness of workflow procedures (van der Aalst, 1998). CPN provide a reasonable level of detail and the representation makes it easy to understand how different processes interact with each other.

11. ***Architecture of Integrated Information Systems (ARIS)*** – ARIS represents an information system for supporting business processes from different views and across all phases of development (Scheer, 1998). The views are:

    - Data – formed by conditions and events.
    - Function – describes the function and its relationships.
    - Organisation – describes the structure and the relationships between users and organisational units.
    - Resource – information technology components.

    By dividing the model into these individual views, the complexity is reduced, but the relationships between the views is lost – thus, the need for another view, the 'control view', which retains the relationships (Scheer, 2004). Figure 7 shows an ARIS model of a customer complaint process provided by the industrial sponsors:

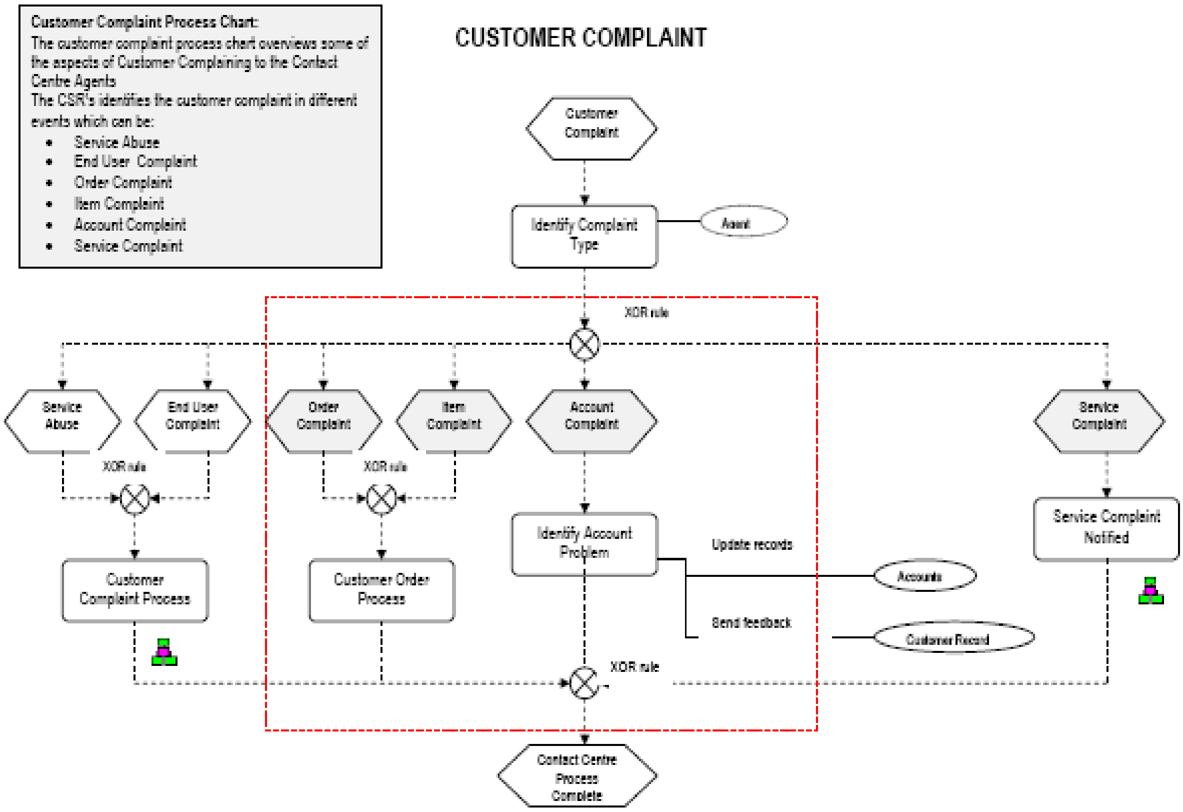


Figure 7: ARIS Process Model (as supplied by industrial sponsor)

12. ***Workflow/Workflow Management Systems (WFMS)*** - Workflow is a flow of tasks between computer applications or people in an organisation. Two or more members of a workgroup, in order to reach a common goal, can define workflow, as well as any tasks performed, in series or in parallel. Workflow is more than a technique to model a process - it is a method to analyse and improve a process, including its modelling. As such, it is associated with BPR. It is defined as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" (Workflow Management Coalition (WfMC), 1997). One minor problem with this technique is that there is no particular notation for workflow systems. This is due to the existence of a number of workflow languages which aim to describe and to specify workflow. Each one of these languages uses a specific notation, sometimes a graphical one, to describe the processes (Aguilar-Saven, 2004). The first attempt at standardisation in workflow was made by the Workflow Management Coalition in 1993.

WFMS range from simple, email based tools to very sophisticated integrated development tools which can be used across entire organisations (Kampf and Grobmann, 2006). WFMS is described as "a system that defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the business process logic" (Workflow Management Coalition (WfMC), 1997). One example of a computer representation of business process logic, or a workflow language, is JPDL, the process language used by jBPM, a WFMS developed by JBoss. Figure 8 shows a jBPM model:

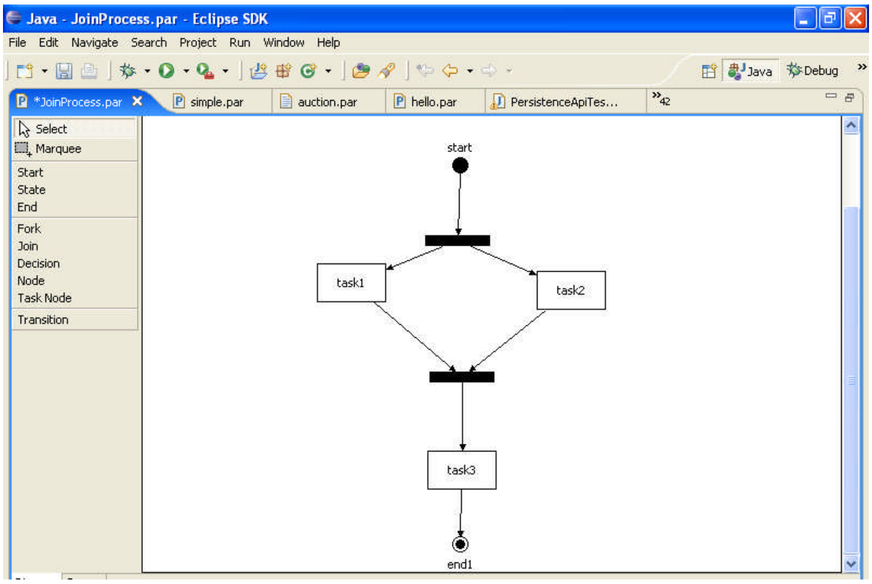


Figure 8: Business Process Model in JBoss jBPM

Another example is process modelling using ARIS, which allows the modeller to define individual tasks and interdependencies, and then its translation to process language, BPML - this then allows execution of the process by a workflow engine (Scheer, 2004). There are a large number of WFMS available on the market, including: OpenEbXML, Werkflow, OSWorkflow, wfmOpen, OFBiz, ObjectWeb, Bonita, Bigbross Bossa, XFlow, Taverna, Enhydra Shark, PowerFolder, Breeze, Open Business Engine, OpenWFE, Freefluo, ZBuilder, Twister and Con:cern (Baeyens, 2004).

WFMS also allow storage of process execution data for analysis. However, workflow techniques are developed to allow the automated execution of processes rather than the detailed modelling of a process – the model contains sufficient information for a workflow engine to execute the process in the desired sequence, and may even be linked to targets and performance indicators to facilitate improvements in its execution, but complexity of processes is again an issue, along with sufficient information being available to allow comparisons of the process model with alternative process data.

Table 1, shows a cross section of available modelling techniques, some of which have been reviewed in this section. Their attributes, characteristics, and strengths and weaknesses from both a user and modeller perspective, are listed.

Table 1: Business Process Modelling Techniques (Aguilar-Saven, 2004)

Technique	Description	Attributes	Characteristics	Strengths and Weakness			
				User perspective		Modeller perspective	
				Strength	Weakness	Strength	Weakness
Flow Chart	Graphic representation	Flow of actions	Not sub-layers Great details No overview	Communication ability	Can be too large	Flexibility quick, simple	No method available Different notations
DFD	Descriptive diagrams for structured analysis	Flow of data	Explains logical level sub-layers	Easy to understand	Only flow of data is shown	Easy to verify and draw	
RAD	Graphic view object state transition diagrams	Flow of individual roles	Detailed view Degree of empowerment No overview	Supports communication Intuitive to read	Not possible to be decomposed	Include business objects	Different notations
RID	Matrix representation of processes for co-ordination of activities	Flows of activities and roles	Inputs to and outputs from are not modelled Performers are included	Intuitive to understand	Important information is not included	Rigid notation Complex processes can be displayed	Difficult to edit an existing diagram Hard to construct
Gantt Chart	Matrix representation	Flow of activities and duration	Relate activities to time	Easy overview representation and control of performance	Not aid for analysis or design	Simple	No clear representation of dependencies
IDEF0	Structural graphical representation, text and glossary	Flows of activities, inputs, outputs, control and mechanisms	Based on SADT Sub-layers The most popular	Shows inputs, outputs, control and mechanisms overview and details	Trend to be interpreted only as a sequence of activities Roles are not represented	Strict rules Possible to build a software Quick mapping	
IDEF3	Behavioural aspects of a system	Precedence and causality relationships between activities	Allows different views Process flow descriptions and object state transition description diagrams Sub-layers	Easy to understand dynamic aspects in a static way	Many partial diagrams to describe a process	Strict rules and notation Possible to build a software	Need lot of data Time consuming when modelling complex systems

Table 1: Business Process Modelling Techniques (cont.)

Coloured Petri Nets	Graphical oriented language to design, specify, simulate and verify systems	Network of places, transitions and arcs	Extended Petri nets Token are differentiated by colours Hierarchical decomposition	Easy to understand how individual processes interact with each other	Models are excessively large	Formal mathematical representation Well defined syntax and semantics Possible to build a software	Time consuming when modelling
Object Oriented Methods	Describe a system with different type of objects	Object's structure and behaviour	Three concepts: objects, classes and messages There are many modelling techniques based on OO	Enactable model to control and monitor processes	Model are excessively large and detailed Fragmented information	Data concepts Internal consistency across design, analysis and programming Possible to build a software	Need lot of data Time consuming when modelling Complexity
Workflow	Computerised facilitation or automation of a business process	Flow of information, tasks and procedural rules	Flow of tasks between computers and people Decentralised	Easy to analyse Shorter learning time		Possible build a software Data transfer Easy to make changes	Lack of a particular notation Many languages
Rich Pictures	Contextual representation of things	Represent process human problematic	Represent some of the richness of the process being examined	Support communication and understanding of the process	It is not structured approach	Easy to illustrate components as clients, people, tasks and environment	Lack of a particular notation
GRAI grid and GRAI nets	Descriptive diagrams of the process focused on decisions	Decision making process and flow of activities	Sub-layers Distinction between period and event driven activities	Shows inputs, outputs, control and mechanisms, overview and details	Many partial diagrams to describe a process	Strict rules and notation Possible to build a software	Need lot of data Time consuming when modelling Complexity

Based on the modelling criteria set for this research, none of the techniques reviewed in this section and summarised in Table 1 fully provide all the necessary features for the desired representation. Six of the techniques, IDEF0, IDEF3, CPN, Object Oriented (i.e. UML), Workflow/WFMS and GRAI, possess the strength of “possible to build a software”, but none individually fulfil all of the necessary criteria for use in this research. However, Workflow is the closest technique available which fulfils a large number of the set criteria for the following reasons:

- It is relatively easy to understand and change without specialised knowledge.
- It allows the computerised facilitation or automation of a business process execution.
- It represents the flow of tasks between computers and people, along with information and procedural rules.
- It enables storage of process execution data

2.2. Business Process Reengineering

There is a vast amount of published literature concerning business process reengineering (BPR) and its application. Much of the literature reiterates the principles of BPR and, as such, it was decided to select the key points in this section in order to provide an overview of BPR, with particular reference to the well-known work of Hammer and Champy. The stages and levels of BPR are then reviewed to demonstrate the context of this research in the domain of BPR.

As a result of the amount of literature concerning BPR, there are many different definitions, three of which have been selected as follows:

- “a fundamental re-evaluation/redesign of a company’s business processes and organisational structures in order to achieve dramatic improvements in its critical success factors – quality, productivity, customer satisfaction and time to market etc.” (Tapscott and Caston, 1993).

- “the redesign and improvement of business processes…both in depth…roles and responsibilities, measurements and incentives, organisational structure, information technology, shared values and skills…and breadth…which can lead to long-term profits" (Hall et al, 1994).

- "the fundamental rethinking and redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance such as cost, quality, service and speed" (Hammer and Champy, 1993).

BPR, or Business Reengineering, or Business Process Redesign was developed to enable businesses to become "process driven and customer-focussed" (Cameron and Braiden, 2004), and the significant benefits that BPR can achieve first became apparent in the mid-1980s. BPR is normally undertaken as an organisation wide improvement operation, carried out at a number of different levels, and is applicable to all kinds of organisations, regardless of the type of business they are involved with. Reengineering may be undertaken by individuals, but generally by teams, such as 'Case' teams (permanent) or 'Self-Directed Work Teams' (temporary) (Hammer and Champy, 1993), (Choudrie et al, 2002). The aim is not only to improve the efficiency of the processes the business operates to produce the products or services it offers, but also to improve all activities and structures within the business. For example:

- Introducing cellular manufacturing and/or group technology (see Section 2.6.2) within a manufacturing company in order to facilitate the flow of production.
- Restructuring the layout of offices and the proximity of personnel with linked duties within a service based company to facilitate the flow of service provision.
- Reassigning and/or retraining employees to increase their efficiency.
- Allocating responsibility, or ownership, of processes to suitable individuals or teams of employees.

However, BPR does not always achieve any of the dramatic results intended – in as much as approximately 50% to 70% of cases according to Hammer and Champy (Hammer and Champy, 1993).

BPR is one of a number of improvement philosophies available to businesses, such as Total Quality Management, Just-In-Time, Simultaneous Engineering and Time Compression Management (Burke and Peppard, 1994).

BPR has always been facilitated by the use of IT – this can be described as an 'essential enabler' (Hammer and Champy, 1993). The automation of BPR is not a new concept. This normally assists and facilitates the BPR procedure in varying degrees and in different areas of the overall procedure, providing modelling and simulation facilities in some cases. For example, Computer Aided Process Reengineering treats processes as systems, and utilises systems analysis techniques for management and improvement of the processes (Hansen, 1994). There is such a wide range of available software tools that methodologies have been developed in order to facilitate the selection of the appropriate tool (Bradley et al, 1995). The following table, Table 2, lists some of the guidelines for selecting BPR tools and techniques:

Table 2: Guidelines for Selecting Tools/Techniques for BPR (Gunasekaran and Kobu, 2002)

Areas to be reengineered	Tools/techniques used in BPR	Description for application
BPR strategies	Decision support system and Knowledge-based models	DSS (decision support systems), AI (artificial intelligence) and ES (expert systems) can be used to select suitable strategies and methods for reengineering business processes. This should be based on a set of performance measures and metrics
Business process system design	Analytical models such as queuing and simulations models	They can be used for system design considering the objective of eliminating non-value adding activities and focusing on value adding activities with suitable changes in organisational structure, information systems and technologies
Project management	PERT/CPM (program evaluation review technique/critical path method) and Flow Charts	They can be used for the implementation of various reengineering processes
Reengineering business process	Activity-based analysis and workflow model including flow chart	They can be used to analyse the process and identify the value and non-value-adding activities in order to achieve a dramatic improvement in business performance. They can also be used for the implementation of reengineering business processes
Design of information system for reengineering business process	Object-oriented models and programming	They facilitate a flexible modelling of organisations. This would help to optimise the organisational structure and information flow with the objective of reducing the production cycle time and hence to reengineered business process system
Understanding of the business process system	IDEF Models, EFQM (European forum for quality management) Models, Petri-net Models	They can be used to model business processes in easy to understand visual forms

This gives one suggested set of guidelines for BPR at different levels. The ‘reengineering business process’ category is of particular interest in this research. It can also be seen that process modelling (notably Workflow) and activity based analysis play an integral part in this category of BPR tools and techniques. As far as BPR tools are concerned, there are a vast number, all assisting the reengineering operations in a variety of ways (as illustrated in Table 2). The following is just a small selection of some of those available:

- Apache
- Bonapart
- Business Design Facility
- Business Improvement Facility
- Caddie
- Ithink
- Vensim
- METIS
- DPA
- oCTAVe Process Manager
- Logic Works' BPwin
- Gensym's ReThink
- Clear Process
- Optima! and Optima! Express
- Extend+BPR
- Cosmo
- Workflow Analyser
- FirstSTEP
- Process Charter
- Business Resource Software
- DTIC
- CROSSFLOW
- ETC
- The Workflow Factory

(University of Toronto Enterprise Integration Laboratory, 2005)

2.2.1. Application of BPR

Burke and Peppard define the principles of BPR as follows:

- Radical change and assumption challenging
- Process and goal orientation
- Organisational restructuring
- Exploiting enabling technologies, such as IT

(Burke and Peppard, 1994)

There are 8 rules that should be followed when carrying out BPR at organisation level:

- “Organise around outcomes, not tasks.
- Have those who use the output of the process perform the process.
- Treat geographically dispersed resources as though they were centralised, creating hybrid centralised/decentralised organisations.
- Link parallel activities in a natural order instead of integrating their results.
- Perform work where it makes most sense, particularly decision-making, information processing, checks and controls, making them part of the process.
- Capture information once and at the source, minimising reconciliation.
- Combine several jobs into one, possibly creating a case manager or case team as a single point of contact.
- Create multiple versions of processes where appropriate.”

(Burke and Peppard, 1994), (Hammer and Champy, 1993)

One methodology for reengineering a company, described as “a general guideline which can be customised to suit most situations”, is illustrated in Figure 9:

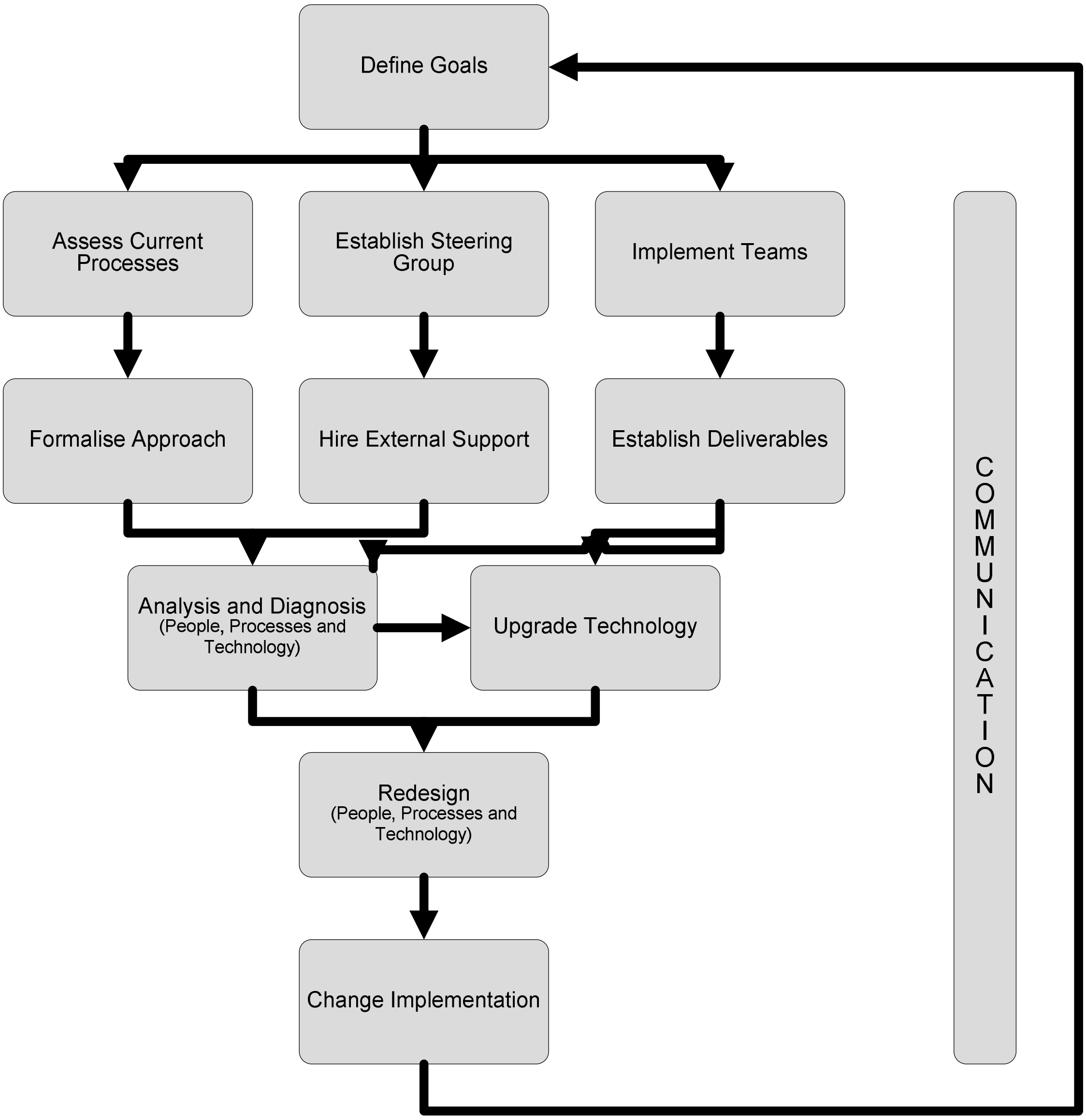


Figure 9: Example BPR Methodology (Towers, 1994)

The methodology comprises of ten steps:

1. Define goals – costs and benefits, time scale, risks.
2. Establish Baseline – assess current processes, form a steering group, implement teams (to undertake analysis and diagnosis).
3. Communication – implement a programme.
4. Formalise approach – techniques, software, project plan detail.
5. Hire external help – may not be necessary.
6. Establish deliverables – key benefits to achieve.

7. Analysis and diagnosis – comparison of ‘as is’ and ‘to be’.
8. Upgrade technology – or redeploy existing.
9. Redesign – start implementation of new organisational structure.
10. Change implementation – deliver, implement and install, test and train.

(Towers, 1994)

In terms of reengineering at different levels of an organisation, rather than the organisation in its entirety, there are four organisational levels to be considered in BPR: people, internal systems and structure, process and objective (Burke and Peppard, 1994). These can be illustrated as follows in Figure 10:

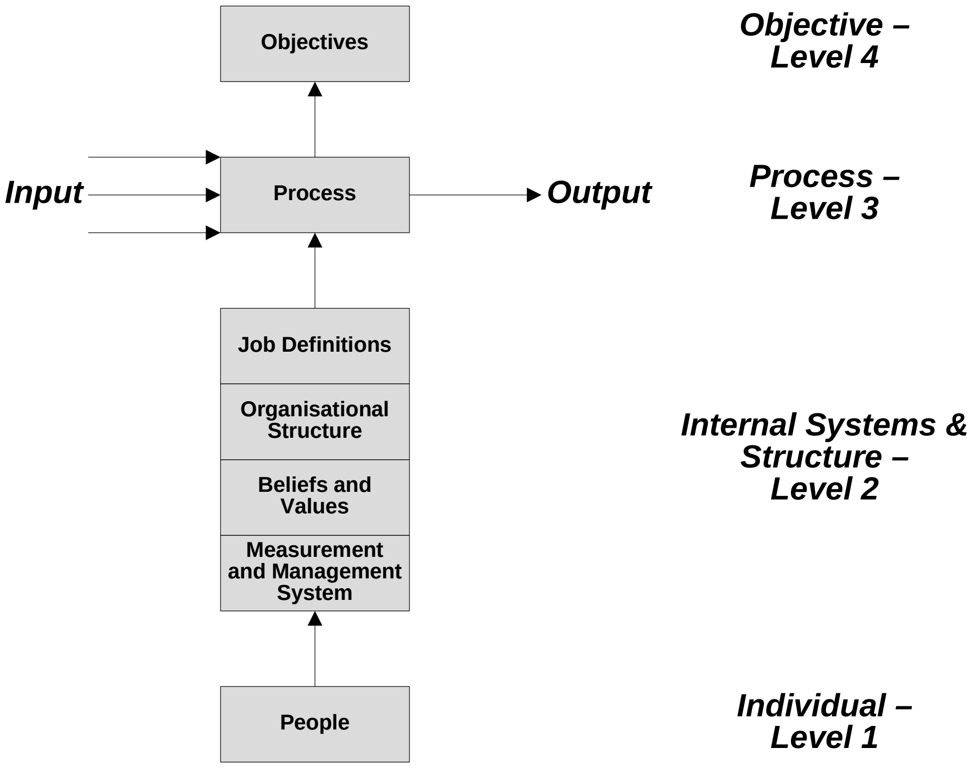


Figure 10: Four Levels of BPR (Burke and Peppard, 1994)

The emphasis in this research is on the automation of BPR at the process and objective levels – specifically, the actual process models the organization executes to provide its services or produce its products, with the objectives of decreasing process cost (Burke and Peppard, 1994) and duration, and maintaining or increasing levels of customer satisfaction (other criteria at the objective level may also include, for example, efficiency and importance (Tan et al, 2007)), are of interest in this research. As such, only a small part of the whole BPR procedure is targeted, but this is an important part as it concerns the low-level processes that form the basis of an organisation's service provision. Most processes are initially designed to be relatively simple, but they can grow complex over time as modifications and additions are made to deal with exceptions or special cases, leading to complicated and inefficient processes (Hammer and Champy, 1993) which require improvements. Also, processes which utilise technology may require updating periodically due to technological advances – the resources available and their capabilities change over time (Grover and Kettinger, 2000). Reengineering at the level below this, Level 2, may involve, for example, the restructuring of office layouts or production/assembly lines. Reengineering at Level l may involve, for example, reallocation of duties and/or staff retraining.

At Level 3, the following rules are suggested:

- Communicate and increase awareness of the process
- Document the process tasks
- Diagram the whole process
- Measure the process

(Hansen, 1994)

These rules can be applied as the preliminary steps to process improvement. However, the actual rules or guidelines which should follow these preliminary steps, and for carrying out the actual reengineering of a process at Level 3, can be determined by asking a number of questions. These are based on BPR at all levels, and can be applied at the process level. They are as follows:

- Does the process contain steps which are repeated or redundant in certain circumstances?
- What would happen if these steps were not completed?
- Are there steps or series of steps which could be performed in parallel or combined with others?

- What are the risks of eliminating repeated or redundant steps or combining actions?
- Are there any problems with certain parts of the existing process which need to be addressed?

(Picot et al, 2004), (Grover and Malhotra, 1997), (Hammer and Champy, 1993)

With the above questions in mind, it is necessary to identify and apply certain guidelines in BPR in order to improve, and maintain the feasibility of, a process. It is one of the purposes of this research to identify and use such guidelines as a part of an automated BPR procedure. These guidelines, or rules, would normally be applied by human experts capable of making decisions concerning the reengineering of the process, invariably assisted by information technology and frequently based on the analysis of process execution logs provided by WFMS and BPMS (Stelling et al, 2008a). All BPR tools and techniques, whatever level they are aimed at, involve human judgement – the tools assist and provide vital information relevant to the reengineering operation (for example, process modelling using Bonapart, simulation and evaluation using Business Design Facility, performance monitoring using Vensim, cost estimating using DPA, value analysis using FirstSTEP) which is then analysed and acted upon. Therefore, an element of automation is present, but the desire in this research is to attempt to remove the human judgement element of BPR at a process model level as much as is possible by significantly increasing the level of automation. The BPR procedure would normally involve the examination of 'as-is' processes, generation and analysis of process data (e.g. cost, duration) by human experts, and then the iterative creation and evaluation of potential 'to-be' processes. This research intends to automate this procedure by automatically generating potentially numerous variations of 'as-is' processes by automating some of the guidelines of BPR at this level – cost estimation, duration estimation and customer satisfaction impact measurement can also be automated in order to evaluate the process variations (i.e. the 'to-be' processes). This can be achieved by analysing all of an organisation's process data with a view to replacing parts of the existing process with alternative process data via an algorithm. This can also be facilitated by representing and examining processes at a task level, rather than the 'activity' level often used by existing tools and techniques – this lower level of detail can allow recognition of potential beneficial changes which otherwise may not be considered.

2.3. Business Process Optimisation

Optimisation is a procedure of finding and comparing feasible solutions until no better solution can be found. A significant portion of research and application considers a single objective although most real-world problems involve more than one objective (Deb, 2001) – for instance, optimising a business process with the objectives of minimising cost and maximising customer satisfaction, as is an aim of this research project. Business process modelling is the basis of business process analysis and optimisation, the purpose of modelling a process not only providing an understanding of how it works but also being a way of identifying any potential improvements in, or problems with, the process and re-engineering the process to implement these improvements or remove these problems. The motivation for organisations is to improve the quality and efficiency of their administrative and production processes and to rapidly and reliably deliver services to businesses and individual customers (Grigori et al, 2004).

The concept of what an optimal business process actually is depends on many criteria and on the aims and objectives of the business involved. The quality of processes is defined by many, often conflicting criteria like costs, duration, or quality of output (Hofacker and Vetschera, 2001). It is important that a business optimisation approach clearly defines and specifies how optimisation is perceived and which aspect or aspects of the process are to be optimised (Lindsay et al, 2003).

There are very few optimisation techniques to be found in literature which are suitable for application to business processes. Most optimisation is carried out manually without the use of any automated techniques (Vergidis et al, 2007). This is primarily due to the qualitative nature of many process representations which makes analysis difficult (Vergidis, 2005). As a result, there is a lack of algorithmic approaches to the optimisation of business processes (Vergidis et al, 2007).

Ideally, BPO should involve the quantitative analysis of a business process model and the re-engineering of the process based on the specified objectives, automated by a tool if possible. Reviewing the quantitative analysis of and potential optimisation of business processes is the purpose of this section - it relates directly to business process modelling and some of the techniques mentioned previously in this chapter. Some of the very few available approaches to business process quantitative analysis and optimisation are overviewed below.

1. ***The Fuzzy Logic Approach*** - as mentioned in Section 2.1.1, the main purpose of this modelling technique is to remove the uncertain and incomplete information of process variables and the major benefit of this technique is the ability to quantify outputs from a process which allows for a process to be tested using different values (Zakarian, 2001). This facility to test a process based on different scenarios can allow the process to be optimised, but not in a methodical way – this must be done by the business analyst and relies on his/her skill in analysing and identifying the optimum process via comparisons of test results.

2. ***Mathematical Programming Formulation*** - this technique considers an additive objective function and a problem in which there are only information resources. A number of constraints are used to depict the process and cover all of its elements. The objective function is minimised or maximised according to the optimisation aim. The main concepts used in the process design are activities and resources. The results of experiments conducted to test this technique suggest that the mathematical approach produced satisfactory results but execution times were poor (Hofacker and Vetschera, 2001), (Vergidis, 2005).

3. ***Quantitative Analysis of RAD*** - This approach focuses essentially upon applying measures based on simple counts of actions and role-coupling. These counts are taken from Role Activity Diagrams (see Section 2.1.1) in an attempt to quantitatively analyse the process model (Phalp and Shepperd, 2000).

4. ***Direct Branch and Bound*** - another analysis and optimisation technique which is arguably easy to implement. The branch and bound algorithm can be stopped during execution when the potential improvement is small, without solving the 'problem' to full optimality – this is possible because a bound on the optimal objective is available. For the process design problem, a branching step consists in adding another activity to the process design under construction. Two strategies are identified to restrict the set of activities considered for branching, forward search and backward search. In experiments carried out by Hofacker and Vetscera, the direct branch and bound algorithm turned out to be considerably faster than the mathematical programming formulation overviewed earlier and also yielded better results (Hofacker and Vetschera, 2001).

2.3.1. Single and Multi-Objective Optimisation

There are essentially two types of optimisation: single objective and multi-objective. When an optimisation problem involves only one objective function, the task of finding the optimal solution is called 'single-objective optimisation' (SOO). This involves searching for a single 'best' solution. When more than one objective function is involved, it is called 'multi-objective optimisation' (MOO). The condition, 'Pareto-optimal', is mentioned regularly in MOO techniques when describing a solution, and can be defined as follows: briefly, in a situation where there are a number of solutions for more than one objective, one solution may be better than another for all objectives or one solution may be better for one objective but worse for another objective – superiority of one solution over the other cannot be established bearing all objectives in mind. The aim is to find good compromises rather than a single solution. This type of solution is known as a Pareto-optimal solution.

2.3.2. Optimisation Algorithms

As this research project is based on the optimisation of service based business processes, the main objectives being low cost and duration and maintenance of or increase in levels of customer satisfaction, it is necessary to overview some of the methods used in single and multi-objective optimisation, both classical and evolutionary, and consider their suitability for application to BPO.

Classical multi-objective methods have existed for over 50 years now and, during this period, many algorithms have been suggested. They can be separated into 4 classes (Hwang and Masud, 1979), (Miettinen, 1999):

1. ***No-preference methods*** – these do not assume any information about the importance of objectives, but a heuristic is used to find a single optimal solution.

2. ***Posteriori methods*** – these use preference information of each objective and iteratively generate a set of Pareto-optimal solutions.

3. ***A priori methods*** – these use information about the preferences of objectives and usually find one preferred Pareto-optimal solution.

4. ***Interactive methods*** – these use preference information progressively during the optimisation process.

The following classical methods, briefly overviewed, use preference information in increasing degrees.

1. ***Weighted Sum Method*** - this method scalarises a set of objectives into a single objective by pre-multiplying each objective with a user-supplied weight. It is probably the simplest of the classical approaches towards solving a multi-objective optimisation problem and is intuitive and easy to use (Deb, 2001).

2. ***Benson's Method*** - this procedure has similarities with the Weighted Sum method, the main difference being that the reference solution is taken as a feasible non-Pareto-optimal solution. A solution is randomly chosen from the feasible region and, thereafter, the non-negative difference of each objective is calculated and their sum is maximised. (Deb, 2001).

3. ***Value Function Method*** - mainly used in practice for multi-attribute decision analysis problems with a discrete set of feasible solutions (Keeney and Raiffa, 1976), although it can also be used in continuous search spaces. The user provides a mathematical value function which relates to the objectives. This function must be valid over the entire feasible search space. By changing the parameters involved in the value function, different Pareto-optimal solutions can be found. This method is quite simple and works well if adequate value function information is available. (Deb, 2001).

4. ***Interactive Method*** - there are a number of interactive methods in existence. With these methods, there is no requirement to know a value function relating to the objectives before starting to solve the problem, unlike with the value function method. As and when some Pareto-optimal solutions are found, their location and interactions are analysed. The main aspect of these methods is the involvement of a decision maker – intermittently during the optimisation process, he/she must provide information about the direction of search, weight vector, reference points, and so on. (Deb, 2001).

Evolutionary methods are different from classical search and optimisation procedures in a variety of ways – for instance, finding and maintaining multiple solutions in one single simulation run is a unique feature of evolutionary optimisation techniques (Deb, 2001). Classical optimisation algorithms are designed to solve a specific type of problem and may not even be applicable to another type of problem. Evolutionary algorithms (EA), which are computer-based problem solving systems that mimic natural evolutionary principles to constitute search and optimisation procedures, are more flexible and require less problem information. Four types of EA are overviewed below.

1. ***Genetic Algorithms (GA)*** - search and optimisation procedures that are motivated by the principles of natural genetics and natural selection. Some fundamental ideas of genetics are borrowed and used artificially to construct search algorithms that are robust and require minimal problem information (Deb, 2001). For many years now, GA have been used extensively in a variety of problem domains, including the

sciences, commerce and engineering - engineering applications are by far the most common, and then, to a lesser extent, industrial applications such as scheduling (Coello Coello et al, 2002). The main reasons for their success are their broad applicability, ease of use and global perspective. GA are different from more normal optimisation and search procedures in four ways:

- GA work via a coding of the parameter set and not the parameters themselves.
- GA search from a population of points and not from a single point.
- GA use objective function information and not derivatives of other auxiliary knowledge.
- GA use probabilistic transition rules and not deterministic rules.

(Goldberg, 1989).

Two major advantages of GA are that they can consider several solutions in parallel and that they can easily be extended to multiple criteria. This would suggest that they could be well suited to BPO.

In order to find an optimal solution, a GA imitates the process of natural evolution. It works on a large number of solutions in parallel, where each solution corresponds to an individual in a population (Hofacker and Vetschera, 2001). GA generate each individual from an encoded string known as a 'chromosome' – these chromosomes are combined or mutated to breed new individuals (Kusiak, 2000). Each solution is represented by a chromosome string. A mutation operation changes the values of randomly chosen positions of that string. The resulting mutated individuals are then selected for mating. A crossover operation exchanges information between two individuals. In a last step, superior solutions are randomly selected to form the new generation. The selection probability depends on an objective function, which defines the optimality condition, and on a 'fitness function', which measures how suitable the selection is based on the objective function definition. This process continues until some pre-defined termination criteria are fulfilled.

GA are particularly effective in dealing with problems where traditional optimisation techniques fail, either due to the irregular structure of the search space (for example, absence of gradient information) or because the search becomes computationally intractable (McCall, 2005).

Single Objective GA deal with one objective value and search for a single point rather than a set of pareto-optimal points. As such, they find the 'best' solution rather than a number of compromise solutions. Multi Objective GA deal with non-dominated objective values in which the improvement of one value has an adverse effect on another value. As a result, they generate a set of compromise solutions, or a set of pareto-optimal points (or a pareto front). Examples are Non-Dominated Sorting Genetic Algorithm (NSGA) and Strength Pareto Evolutionary Algorithm (SPEA)

There are an almost unlimited number of potential GA applications as they can be applied to the vast majority of problems which require an optimal solution – any problem which requires a search for an optimal solution could benefit from GA. As such, the optimisation of service based business processes would be a suitable application.

2. ***Genetic Programming (GP)*** - the principal difference between GP and GA is in the way the solution is represented: GP creates computer programs in the form of computer languages as the solution whereas GA creates a string of numbers (the chromosome) that represent the solution. Essentially, GP consists of four steps:

    - Random generation of an initial population of computer programs composed of functions and terminals appropriate to the problem domain.
    - Execution of each program in the population and assignment of a fitness value according to how well it solves the problem (the 'fitness measure').
    - Creation of a new population of computer programs:
        - Copying of the best existing programs.
        - Creation of new computer programs by mutation.

    - Creation of new computer programs by crossover.
- Designation of the best computer program that appeared in any generation (the best-so-far solution) as the result produced by genetic programming.

(Koza, 1992), (Kusiak, 2000)

3. ***Particle Swarm Optimisation (PSO)*** - PSO is another evolutionary computation technique which was inspired by the behaviour of flocking birds and schooling fish. PSO uses a "population" of particles that move, or swarm, through the problem hyperspace with given velocities. At each iteration, the velocities of each particle are randomly adjusted according to the historical best position for the particle itself and the neighbourhood – the best position, as with GA, is determined via a fitness function. The movement of each particle naturally evolves to an optimal solution. (del Valle et al, 2008)

4. ***Simulated Annealing (SA)*** – SA was motivated by "an analogy to the statistical mechanics of annealing in solids". SA techniques use an analogous set of "controlled cooling" operations for non-physical optimization problems. They transform poor and unordered solution into highly optimized and desirable solutions, and provide the possibility of reshaping mathematical insights from the area of physics into insights for real optimisation problems (Rutenbar, 1989).

A comparison of the popularity of some of the algorithmic approaches overviewed in this section was conducted, based on number of publications from 1969 to 2008. The results were as follows in Figure 11:

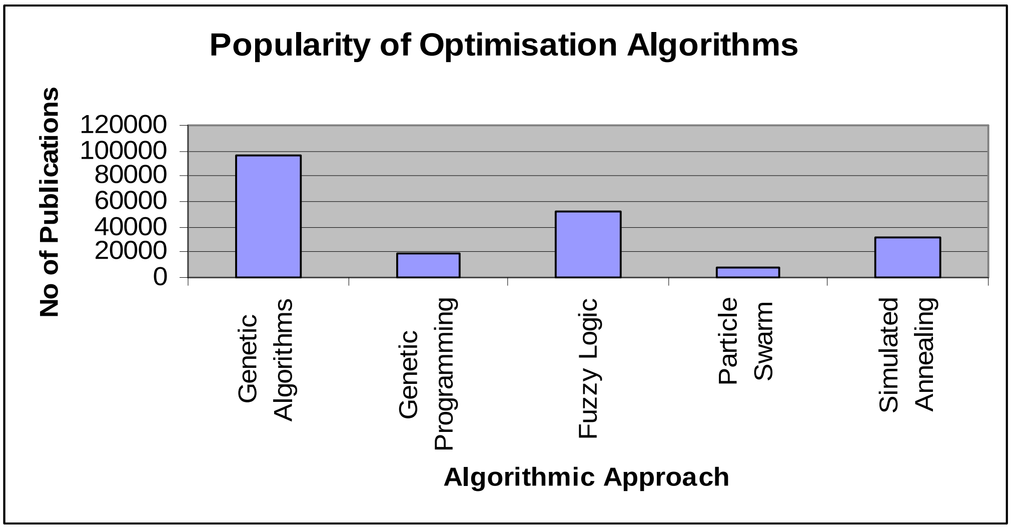


Figure 11: Popularity Comparison of Optimisation Algorithms (Engineering Village, 2008)

As can be seen in Figure 11, GA has by far the most publications, followed by Fuzzy Logic, Simulated Annealing, Genetic Programming and finally Particle Swarm Optimisation.

2.4. Cost Estimation/Modelling

A significant part of this research project is involved with service based process costs and the need to incorporate a cost estimation element into BPR. As a business process is essentially an activity or a series of activities, the emphasis in this section is on Activity Based Costing (ABC), but it also briefly overviews some of the other techniques used to predict costs to demonstrate their unsuitability for this project. It begins with a description of cost estimating.

2.4.1. Cost Estimating

Cost estimating is the prediction of the product cost before its actual manufacture (Aderobal, 1997), (Hackney, 1965) or the process of predicting or forecasting the cost of a work activity or work output (Stewart et al, 1995). The Association for the Advancement of Cost Engineering (AACE) defines cost estimating as "the determination of quantity and the predicting, or forecasting, within a defined scope of the costs required to construct and equip a facility, to manufacture goods, or to furnish a service. Included in these costs are assessments and an evaluation of risks and uncertainties" (AACE, 1990). A good estimation needs to be accurate, reliable, efficient, and transparent – the estimate will need to be justified. The data used for estimating must be relevant, appropriate and of good quality and value (Smith and Mandakovic, 1985).

The cost estimating fundamentals are as follows (Roy, 2003b):

- Understand customer and product requirements.
- Develop breakdown structures (e.g. product, work and process).
- Retrieve and organise (normalise) historical data.
- Develop and use cost estimating relationships.
- Develop and use production learning curves.
- Identify skill categories, skill levels, and labour rates.
- Develop labour hour and material estimates.
- Develop overhead and administrative costs.
- Apply inflation and escalation growth factors.
- Compute the estimate.
- Analyse, adjust, and support the estimate.
- Publish and present the estimate.

2.4.2. Cost Estimating Techniques

There are a number of available estimating methods:

1. ***Parametric*** - this is a common method of estimating cost in the early stages of a system's life cycle when only a few key pieces of data are known (NASA, 2002a). The estimates are based on historical data and mathematical expressions relating cost as the dependent variable to independent cost-driving variables (physical or performance measures associated with the product) through various statistical techniques such as regression analysis (Bashir and Thomson, 2001), (NASA,

2002a). The implicit assumption of this approach is that the same forces that affected cost in the past will affect cost in the future (NASA, 2002b).

2. ***Feature Based*** - a major advantage of Feature Based Cost Estimation is that it can be directly linked to the design process. All components have features attributed to them, and so a total cost can be accumulated as the product is designed if a cost can be related to each particular feature. The more features a product has, the more manufacturing and planning it will require (Brimson, 1998). The inclusion or omission of one specific product feature impacts the downstream costs of a part and could influence the life cycle costs of the product. The major flaw with the process is that it is very difficult to define and then categorise features (Sivaloganathan et al, 1998): different engineering activities describe features in different ways.

3. ***Analogous*** - this technique assumes that similar products or programmes have similar costs (Roy, 2003a). Analogous estimates are performed by reusing and analysing actual data from completed projects, previously stored in a database, to estimate the proposed project (NASA, 2002b), (Roy, 2003a), (Shepperd and Schofield, 1997). These previous costs are applied to the proposed project, although the differences between the proposed project and previous projects it is based on have to be quantified (Baker, 2001). Analogous estimation can also be used when the cost for a generic product, with very little definition, needs to be estimated.

4. ***Activity Based*** - ABC is an accounting technique that allows an organisation to determine the actual cost per product or service regardless of the organisational structure (Office of Information Technology (OIT), 2005). It can give an organisation a competitive advantage through a greater understanding of its product costs, service costs, process costs and the organisation's overall cost behaviour – according to Cokins, it can be thought of as the mathematics used to reassign costs accurately to cost objects (outputs, products, services and customers) and its primary purpose is for profitability analysis (Cokins, 1996).

ABC began as a solution to the problems associated with traditional indirect expense and overhead cost allocation practices. Developed in the late 1980s, it is an activity cost measurement process based upon historical, observed or estimated data (Cokins, 1998), (Börjesson, 1994), (Dean, 2003), (Upchurch, 2002), (Zhang, 1996). It works by taking costs and associating them with the activities that made them necessary and then by taking the accumulated cost of the various activities and associating them with the jobs, products or services that made the activities necessary. Figure 12 is an illustration of the basic steps involved in ABC.

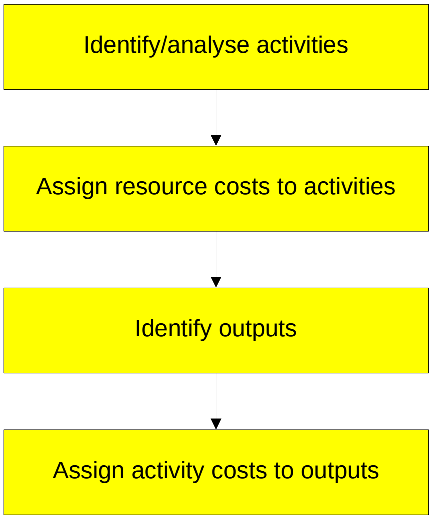

Identify/analyse activities
Assign resource costs to activities
Identify outputs
Assign activity costs to outputs

Figure 12: ABC Steps

To be effective, a company's ABC model must:

- Begin with costs that are accurately defined and measured.
- Properly reflect the cause and effect relationships between the company's product/services, activities and costs.
- Be used in appropriate ways.

(Hicks, 1999)

The methodology is not applied during the conceptual phase of the project development because it needs a deep understanding of the product definition. It helps to accurately identify the costs and performance of activities (administration and manufacturing processes), resources, and cost objects. It principally helps allocate overhead costs (Aderobal, 1997), (Innes et al, 1994), (Kline, 2003) and aids decision-making by increasing costing awareness and cost assignment accuracy (Börjesson, 1994), (Dickinson and Lere, 2003).

Organisational activities first have to be identified and an average cost associated to each of them. Estimating the amount of activity a product is likely to require so that the relative costs can be associated is the next phase (Chalos, 1991). ABC handles the allocation of costs that cannot be directly attributable to a particular product (indirect expenses).

This technique combines both estimates and hard data and consequently is used by most companies (Roy, 2003a). Improved cost management and control, allocation of overheads, and more accurate cost information and product costing are some of the benefits gained by companies that have adopted ABC (Innes et al, 1994). However, it is a highly detailed and complex system and so the technique becomes difficult to maintain (Horsch, 1998).

ABC is most useful to a company where:

- Overheads are high.
- Products are diverse in terms of complexity, volume and amount of direct labour.
- Cost of errors is high.
- Competition is intense.

It is least effective in a company when:

- It has very similar products/services.
- Overheads are low.
- There are homogenous production/conversion processes.
- It has homogenous customers, customer demands and marketing channels.
- Selling/distribution/administration costs are low.

- Very high margins.

(Roztocki, 2005)

ABC was at its most popular in the late 1980's but by the mid-1990's lost its initial impact as companies focused more on growth. However, it has been making a comeback for some years as companies now focus on profitability (Aliga and Sarmenta, 2001). Many consultancies offer ABC/ABM (activity based management) services, and ABC modules form a part of most Enterprise Resource Planning (ERP) systems (SAP for example).

Two of the many uses of ABC, according to Cokins, are for measuring the costs of business processes and for reengineering and process improvement (Cokins, 2001) – both of direct relevance to this project.

2.4.3. Cost Estimating Techniques Summary

Table 3 briefly describes the main characteristics of the cost estimating techniques described in Section 2.4.2, and indicates their principal applications:

Table 3: Cost Estimating Techniques Summary

Technique	Characteristics	Applications
Parametric	Uses historical data	Product and system design/development
Feature Based	Cost relates to features used in design	Product and system design/development
Analogous	Reuses and analyses data from completed projects	Product and system design/development
Activity Based	Can determine cost of product/service regardless of organisational structure	Business Processes, reengineering and improvement

It can be seen in Table 3, that parametric, feature based and analogous cost estimating is generally applied to product and system design and development projects - for instance, designing and developing a new car, computer system or building. They may also be long term projects of many years' duration. As such, these techniques differ considerably to ABC – business processes do not involve the design and development of products or systems; they generate a service via a series of activities which have an operational cost attached to them through the application of ABC. ABC is used, for example, in logistics, banking and telecommunications. However, it is certainly possible to employ elements of the different techniques in combination where appropriate – for instance, analogous for estimating the cost of a new business process which has many similarities to an existing process. The results of a survey on cost estimating in the service sector are presented in Chapter 6, and give an indication of how cost estimating, notably ABC, is applied to service sector business processes.

2.5. Customer Satisfaction

According to Cokins, the achievement of value for customers is now a priority for managers, and customers, who are gaining in power, are willing to pay a premium for that value (Cokins, 1996). As Carr astutely observes, "every organisation has customers – every one" (Carr, 1990). As such, all organisations must strive to create value for their customers in a way that is noticeably better than the competition (Band, 1991) in order to retain existing customers and attract new ones. Companies that are heavily reliant on satisfied customers for repeat business generally are found to have high customer satisfaction levels, whereas this is not the case for organisations who do not experience much, if any, competition in the marketplace (Fornell, 1992) – an extreme example would be public or local government services, notably a Council Tax department, in which the customers are essentially 'captive' and can only take their business elsewhere if they leave the area - which they are unlikely to do due to poor service experiences (Stelling et al, 2008c). Processes should be constantly monitored using cost, duration, quality of output and measurements of satisfaction, and then adjusted periodically in order to keep up with or stay ahead of the competition (Johnson and Weinstein, 2004).

Customer satisfaction levels are generally measured and monitored via customer feedback – classed as 'soft' measures, rather than the 'hard' measures of quantity, size, time etc. (Hayes, 1998).

There are variations in the format of customer feedback: the structure of the questionnaire and categories included are dictated by the priorities of the organisation and the nature of its business – the product or service that they sell. An example of this type of structure, taken from a survey that took place in two different branches of the Commercial Bank of Greece, is shown below in Figure 13:

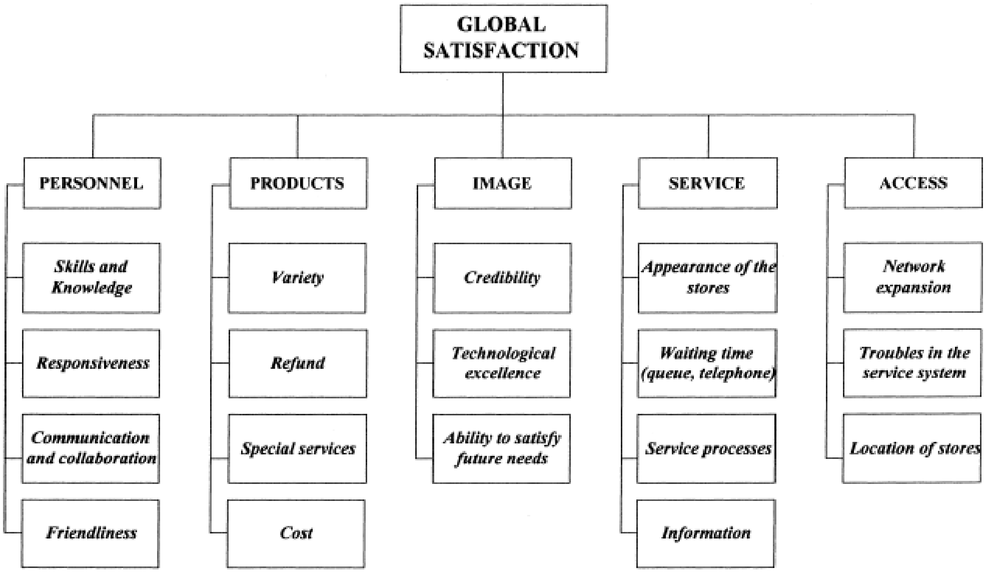


Figure 13: Example Customer Satisfaction Survey Structure (Mihelis et al, 2001)

However, certain aspects of customer satisfaction can be measured without the need to collect opinions from customers – there are some elements of a service experience which have a negative effect on virtually all customers and that apply across the board. Some examples are: being put on hold when making telephone calls; numerous telephone menus to work through; and mistakes made, perhaps leading to repetitions of parts of the process (Gerson, 1993). When and if these negative examples are removed from the business process, a positive effect can occur.

2.6. Classification Techniques

Three areas which fit under the title of ‘Classification Techniques’ have been reviewed in order to provide a potential foundation for a process representation technique.

2.6.1. Classification and Coding Systems

Many organisations have developed their own classification and coding systems. This applies not only to the manufacturing and production industries which use classification and coding as a part of their group technology structure, but also to many record keeping organisations who hold vast numbers of diverse records.

According to Library and Archives Canada (Information Management Services), a classification system is a key foundational element of Information Management within an institution and is a roadmap that provides the rules and definitions used to store and retrieve business information (Library and Archives Canada (LAC) - Information Management Services, 2004).

The structure of a classification system is usually hierarchical and reflects the analytical process as follows:

- first level – the business ***function***
- second level - ***activities*** constituting the function
- third level – groups of ***transactions*** that take place within each activity

(National Archives of Australia, 2000)

This structure is illustrated in Figure 14.

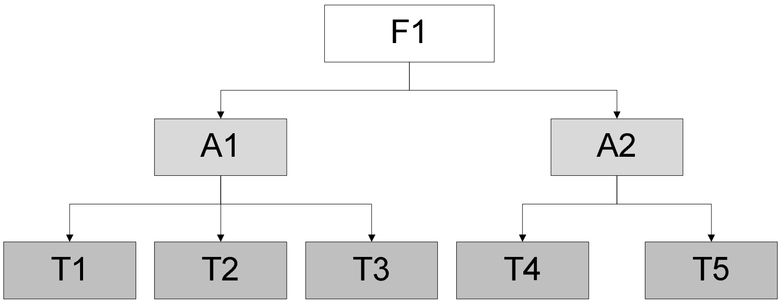


Figure 14: Suggested Classification and Coding Hierarchy

Many different types of coding systems exist in industry. Most share similar characteristics and follow similar principles. According to Hyde, simplification is the most important aspect of a coding system, very closely followed by standardisation (Hyde, 1981). These principles include the following:

- The shorter the code, the fewer the errors
- Identifying codes should be of fixed length and pattern
- Purely numeric codes produce less errors than alphanumeric combinations, but these are acceptable if the alphabetic field is fixed and used to break a string of digits

Although purely numeric systems are less prone to error, many coding systems use a combination of alphabetic and numeric characters - for instance, FA02 could be used for Finance and Accounting department, business expenses section.

2.6.2. Group Technology

Group technology (GT) is described as a technique for identifying and grouping similar or related components in a process in order to take advantage of the similarities that exist between the components. Some of the advantages of GT in manufacturing are a reduction in work in progress and finished stock levels, space is better utilised, production planning and control is improved and less variety of tools is required (Gallagher and Knight, 1986). Unlike its role in the manufacturing industry, the importance of GT to this research is the method of grouping and then classifying and coding the many different processes that take place in a service based company.

2.6.3. Taxonomies

The word taxonomy, associated principally with biology, is defined as “the science of finding, describing, classifying and naming organisms”, “the branch of science concerned with classification”, and also as “a scheme of classification” (Oxford English Dictionary, 2006), (Wiktionary, 2006). Taxonomies are often hierarchical structures consisting of parent-child relationships but may also refer to relationship schemes other than hierarchies, such as network structures. Taxonomy can be described as the enterprise of classifying and naming, and taxonomies are the results of that enterprise (University of Tennessee, 2006). As such, a taxonomy can be used to depict the organisation of a company’s business processes, breaking down the processes into hierarchical structures beginning with the root node, ‘domain’, underneath which may be multiple categories of ‘activities’ followed by ‘transactions’ or ‘topics’. The basic structure is illustrated in Figure 15.

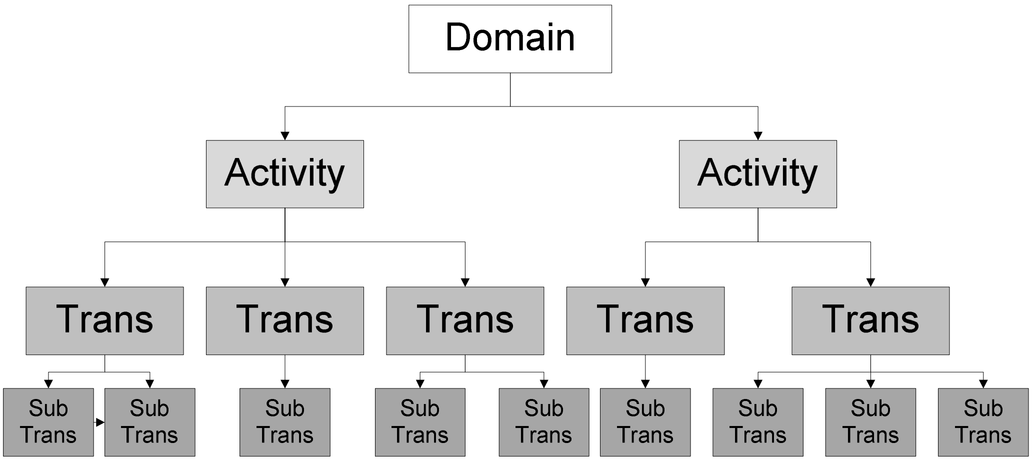


Figure 15: Possible Business Process Taxonomy

This tree structure can be extended further to include sub-levels such as ‘sub-domain’, ‘sub-transaction’, ‘sub-sub-transaction’ and so on, until an acceptable degree of detail is reached (Stelling et al, 2009). This degree of detail and the naming of the levels are dependent on the intended purpose of the taxonomy.

2.7. Relational Database Design

There are a number of techniques available for designing relational databases, generally stemming from work by Chen (Chen, 1976), Codd (Codd, 1990) and Date (Date, 1990). Among these, probably the most popular is the Entity Relationship (ER) model due to its simplicity and ease of understanding without specialised knowledge, first described by Peter Chen in 1976. Briefly, the basic ER model is composed of entities, relationships, attributes, connectivity and existence:

- Entities – the principal objects about which the system needs to store data.
- Relationships – the association between entities
- Attributes – the facts about entities
- Connectivity – describes the occurrences in the relationship, ‘one’ or ‘many’
- Existence – describes whether an entity in a relationship must exist, ‘mandatory or ‘optional’

(Teorey, 1994), (Riordan, 1999)

The following example in Figure 16 shows a one-to-many (connectivity), mandatory (existence) relationship between the entities, ‘salesperson’ and ‘sales’, with ‘name’, ‘number’ and ‘amount’ as attributes of entity ‘sales’:

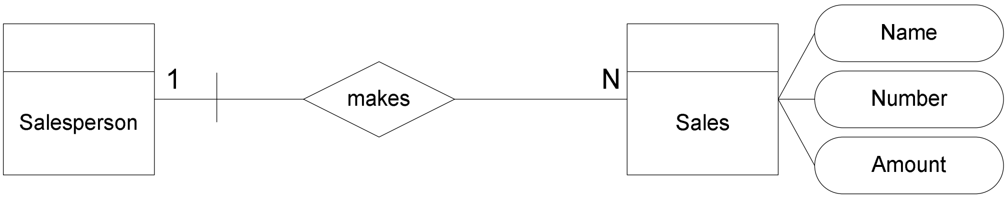


Figure 16: Entity Relationship Diagram

As with process modelling, there is a lack of standardisation in the notation used in ER models – Chen, Crow’s Foot and IDEF1X are three examples, all following the same principles but using different notations.

2.8. Summary of Research Findings

Research findings indicate the following relevant points:

- Process models are mainly diagrammatical in form in order to facilitate ease of understanding. This makes them qualitative in nature and impossible to input to or analyse quantitatively by algorithmic means. Formulaic or mathematical process representations exist which allow input and quantitative analysis, but they cannot represent complex processes, generally catering for serial processes with very few steps. Programmed representations also exist, used by WFMS and BPMS to enable execution of processes, but comprehensive process representation is an issue, along with sufficient information being available to allow comparisons of the process model with potential alternative process data.

- ABC structures are implemented with a degree of flexibility according to organisational requirements, but they share many similarities and follow the basic principles of ABC. However, ABC does not cater for estimating the cost of processes whose component steps may vary in their amount of resource usage due to the outcome of decisions or events during the process execution - this should have been catered for when originally assigning costs to activities and during periodic updates, perhaps based on online transaction processing data, but if a process is to be reengineered and this resource usage would change as a result, it must be detected and measured as it will effect the cost of the process. Also, ABC does not assign costs at task level, but purely at activity level.

- Customer satisfaction is generally measured and monitored via customer feedback or analysis of customer behaviour – the most effective methods for compiling customer data. Situations are not catered for in which an instant judgement is required on a reengineered process’s impact on satisfaction without the availability of customer feedback or process related customer data.

- BPR is carried out at a number of different levels, generally undertaken by human experts capable of using their judgement (with the assistance of Information Technology). In large organisations with data warehouses containing vast amounts of process data, the human effort in maintaining complete process optimality is considerable, if not impossible.

2.9. Summary and Discussion

The summary and discussion is split into three areas of relevance to the project: process modelling; process analysis, reengineering and optimisation; and process evaluation.

2.9.1. Process Modelling

Traditional process modelling techniques share one main characteristic – the diagrammatical representation of a process enabling the user to visualise the flow of activities within the process. Apart from this common characteristic, all traditional techniques differ in their method of depicting the business process. They are generally based on informal notation and lack quantitative information that could enhance their standardisation. They are also mainly high level modelling techniques and as such contain uncertain and incomplete information (Zakarian, 2001). Traditional techniques are also unable to address the complexity and non-linear nature of many business processes within organisations – they tend to concentrate on the internal structure of the process and organisation and assume the existence of crucial notions such as perfect knowledge and the human participants as rational decision makers cooperating together to achieve agreed and clearly defined goals (Lindsay et al, 2003) – not usually the case in reality. However, these techniques allow the modeller to discuss and validate process models with both users and process owners, many of whom are not prepared to invest their time in understanding more complex representations (Phalp and Shepperd, 2000). Therefore, for the above reasons, it must be concluded that no traditional techniques are suitable for use in this project in their entirety – in most cases, they do not provide enough information, and in all cases they cannot be analysed quantitatively by an algorithm.

Parametric or formal modelling techniques aim to address the shortcomings of traditional techniques. In order to make process modelling techniques more attractive, it is argued that formal techniques for analysis of process models are required (Zakarian, 2001). Formal process models need to represent both the dynamic and functional elements of the process (Aguilar-Saven, 2004) and also need to enable quantitative analysis - not just the qualitative analysis offered by traditional methods. The most noticeable aspect of formal techniques compared to traditional techniques is their increased complexity. Whereas traditional techniques are generally easy to use and require no particular peripheral skills, other than an understanding of the process to be modelled, formal techniques involve many constraints and often mathematical and programming expertise. However, these complexities lead to process models which are much more accurate than traditional representations and that can be effectively analysed and verified to a certain level. Although these techniques are certainly more accurate than traditional ones, none of them fulfil all of the required criteria for use in this research project.

It is workflow and WFMS that provide ideas for the required representation of process models in this project. jBOSS jBPM and ARIS, mentioned earlier, allow the graphical modelling of a process, its translation into JPDL and BPML respectively, viewable as an XML type of program code, and its automated execution during which details such as task duration, can be stored in a database. However, these workflow techniques are developed to allow business process automation, the automated execution of processes, rather than the detailed modelling of a process to the extent that every component can be analysed and compared against other available process components in a data warehouse for suitability in reengineering. The model contains sufficient information for a workflow engine to execute the process, but comprehensive process information is again an issue.

In order to fulfil all the criteria for this research, there is no single tool or technique available. It was therefore necessary to use elements of some of the tools and techniques reviewed to create a novel modelling approach. The flow diagram provides a basis for this as it is flexible, easily readable and understandable, and, most importantly, can depict the flow of a process at a desired level of detail. If codes could be used to represent each element of the flow diagram, it could then be translated relatively easily into a quantitative model for input to an algorithm – this idea stems from the conversion of diagrammatic representations into computer code used in WFMS. Review of Group Technology, Taxonomies and Classification and Coding Systems reinforce this proposed method, suggesting a potential implementation strategy and providing guidelines for a coding mechanism. Also, lack of standardisation amongst modelling techniques suggested a requirement to include OMG's Business Process Modelling Notation (BPMN) in the process modelling technique, as OMG are attempting to establish this as a standard and claim to cater for all possible notations.

2.9.2. Process Analysis, Optimisation and Reengineering

The approaches to BPO reviewed all enable varying degrees of quantitative analysis of business process models, with a view to optimisation based on this analysis. They are all based on extracting quantifiable measures from diagrammatical representations of processes, enabling some degree of optimisation according to these measures. However, none of these techniques provide a complete, quantitative picture of a business process – without this information, it is not be possible to utilise these techniques to any great effect.

The classical optimisation methods overviewed all suggest a way of converting a multi-objective optimisation problem into a single-objective optimisation problem. For instance, the weighted sum technique suggests minimising a weighted sum of multiple objectives and the value function technique suggests maximising an overall value function relating all objectives. These conversion techniques result in a single-objective optimisation problem which must then be solved by a single-objective optimisation algorithm. In nearly every case, the optimal solution is expected to be a Pareto-optimal solution – this is specific to the parameters used in the conversion so, in order to find a different Pareto-optimal solution, the parameters must be changed and the resulting new single-objective optimisation problem must be solved again. Therefore, to find, for example, 20 different Pareto-optimal solutions, at least 20 different single-objective optimisation problems must be formed and solved. Also, all of these classical methods require a degree of knowledge about the problem as they involve user-defined parameters which are difficult to set in an arbitrary problem (Deb, 2001). As such, none of these techniques are particularly appropriate for use in this research project – it is not possible to automate any of them and, if it was, they would not be efficient enough.

There are a number of different evolutionary algorithms, of which four of the main ones have been overviewed. Most of these evolutionary methods share a number of common properties:

1. They all work with a population of solutions, instead of one solution in each iteration, as with classical methods. By starting with a random set of solutions, an EA modifies the current population and creates a different population in each iteration. By working with a number of different solutions, an EA is able to capture multiple optimal solutions in one single simulation run.
2. They have two distinct operations – selection and search. During selection, better solutions in the current population are emphasised by duplicating them in the mating pool. During search, new solutions are created by exchanging partial information among solutions of the mating pool.
3. They do not use any gradient information during the selection and search operations. Also, their representations are flexible. These properties make EA flexible enough to be used in a wide variety of problem domains.
4. The operators use stochastic (non-deterministic) principles – since deterministic rules are not used, EA do not assume any particular structure of a problem to be solved.

(Deb, 2001)

The common properties of EA listed above make them more flexible and powerful than any of the classical methods overviewed. They are computerised and can therefore be directly applied to databases and information warehouses to search for optimal solutions. GA, by far the most popular optimisation algorithm, can provide a means of maximising the available search space, and generate and analyse potentially very large numbers of processes – necessary when dealing with large organisations' process data.

These characteristics make EA of interest in this research project. There are three criteria to be considered – cost, duration and customer satisfaction – which appears to make this a multi-objective optimisation situation, but there are links between all the criteria which have an influence on this situation.

BPR aims at organisation-wide improvements and is split into a number of different levels. It is outside the scope of this research to propose an automated methodology for the reengineering of entire organisations. However, reengineering at two of the levels described by Burke and Peppard (Burke and Peppard, 1994), the process and objective levels, are of direct relevance to this research – specifically, the low-level process models which an organisation executes in order to provide its services. At the process level, BPR involves the application of certain guidelines by human experts. Some of these guidelines can be automated and a degree of artificial intelligence incorporated, via an evolutionary based optimisation module, in order to generate feasible, reengineered processes that cater for the specified objectives. Existing BPR techniques and tools facilitate the BPR procedure and, as such, include a degree of automation, but it is the desire of this research to extend the automation as far as possible and to attempt to remove human involvement to a large degree.

2.9.3. Process Evaluation

Due to its link with activities, ABC immediately appeared to be a suitable option for use in this research project: there is a direct association between activities and business processes, as a business process is essentially an activity or a series of activities. The requirement to break down business processes and reengineer them based on cost criteria is entirely compatible with this technique. A structure is required that allows differentiation between reengineered processes that use the same resources as the original process but in different degrees, either due to changes in process duration or changes in probabilities of resource usage, or reengineered processes that may use some different resources that still produce the same output. As such, a probability driven, direct resource-to-task relationship is required.

There are certain aspects of customer satisfaction that can be measured without the need to collect opinions from customers – there are some elements of a service experience which have a negative effect on virtually all customers and that apply across the board. These aspects could be used to develop an automated methodology for detecting impact, either positive or negative, on customer satisfaction.

2.10. Summary of Research Gaps

The research gaps identified are linked to the findings summarised in Section 2.8. They are as follows:

- A requirement for the development of a quantitative complex process representation mechanism which enables input to, analysis and evaluation by an algorithm, and which also contains sufficient information to allow comparisons to be made with all available process data to ascertain its suitability for replacements of parts of existing processes

- A requirement to introduce another level of detail to an ABC estimate when applied to an automated reengineering technique for service sector processes - by developing a probability (of a process following certain paths) driven 'Activity and Task Based Costing' methodology.

- A requirement to develop a methodology which uses available process data to attempt to predict an impact on customer satisfaction.

- A requirement for the development of a complex process reengineering, optimisation and reconfiguration technique with the ability to automate BPR using as much judgement as possible via computerised means, by introducing a degree of artificial intelligence. The technique should utilise cost estimation as the principal evaluation criterion.

The next chapter presents the development of the research aim, objectives and methodology.

3. Research Aim, Objectives and Methodology

This chapter again presents the research aim, plus the objectives which were derived from the research gaps identified in Chapter 2, along with the methodology used to achieve the aim and objectives. It briefly discusses quantitative and qualitative research methods, the research methodology utilised, and the data collection and evaluation methods adopted in order to complete the research.

3.1. Research Aim and Objectives

3.1.1. Aim

The research aim, as mentioned in Chapter 1, is to develop a framework to automate BPR in the service sector using process cost, duration and impact on customer satisfaction as criteria.

3.1.2. Objectives

The key objectives of the project, which relate directly to the research gaps identified in Chapter 2, are to:

1. Develop a mechanism for representing a process model to allow input to an algorithm.

2. Develop an operational cost modelling approach for business processes.

3. Develop a means of detecting impact on customer satisfaction.

4. Develop an automated reengineering plus optimisation approach for business processes using cost, duration and customer satisfaction criteria.

5. Develop a prototype software platform.

6. Evaluate the integrated automated business process reengineering technique.

3.2. Development of a Research Methodology

In order to achieve the aim and objectives of this research, it was necessary for information to be gathered via a number of sources, such as published literature, case studies, and the opinions of experts.

3.2.1. Types of Research

Research generally consists of two basic approaches: the scientific, fixed or quantitative, and the naturalistic, flexible or qualitative (Robson, 2002), (Burns, 2000), (Gummesson, 1991).

1. ***Quantitative Research*** - Quantitative methods are built upon a foundation of premises and beliefs, including the assumption that data must yield proof or strong confirmation of a theory or hypothesis in a research setting. The most important characteristics of scientific research are control, operational definition, replication, and hypothesis testing. Control provides unambiguous answers to identify why something happens, what causes some event, or under what conditions an event occurs. Operational definition means that terms must be defined by the steps or operations used to measure them. Replication means that data obtained in an experiment is reliable, in that identical results must be found if the study is repeated. Hypotheses are systematically created and subject to empirical tests (Burns, 2000). Table 4 illustrates the main strengths and weaknesses of the quantitative approach.

Table 4: Quantitative Approach: Strengths and Weaknesses

	Strengths	Weaknesses
Quantitative Approach	Results are replicable	Removed from everyday life
	Results are verifiable	Difficult to respond to environmental forces
	Offers control and precision	Does not account for people's unique experiences
	Illustrates causal effects	Lacks flexibility

2. ***Qualitative research*** - Qualitative methods always involve some kind of direct encounter with the real world, whether it takes the form of ongoing daily life or interactions with a selected group. The qualitative researcher is concerned not only with objectively measurable 'facts' or 'events', but also with the ways that people construct, interpret, and give meanings to these experiences. Furthermore, qualitative approaches typically include attention to dynamic processes rather than, or in addition to, static categories, and they aim to discover or develop new concepts rather than imposing preconceived categories on the people and events they observe. Qualitative research can provide a more informative picture of culturally based processes and practices, and a depth to context-based explanations of events, processes, outcomes, and ultimately future policy and practice (May, 2002).

   Table 5 illustrates the main strengths and weaknesses of the qualitative approach.

Table 5: Qualitative Approach: Strengths and Weaknesses

	Strengths	Weaknesses
Qualitative Approach	Direct encounter with world	Time required in research setting
	Allows unique experiences to be taken into account	Problems with validity and reliability
	Studies objects in entirety	Problems of anonymity
	Contact with participants	Possible bias

The following table, Table 6, provides a comparison between the two techniques described above.

Table 6: Comparison of Qualitative and Qualitative Research Strategies (Burns, 2000)

Qualitative	Quantitative
Assumptions	
Reality socially constructed	Facts and data have an objective reality
Variables complex and interwoven; difficult to measure	Variables can be measured and identified
Events viewed from informants' perspective	Events viewed from outsiders' perspective
Dynamic quality to life	Static reality to life
Purpose	
Interpretation	Prediction
Contextualisation	Generalisation
Understanding the perspectives of others	Causal explanation
Method	
Data collection using participant observation, unstructured interviews	Testing and measuring
Concludes with hypothesis and grounded theory	Commences with hypothesis and theory
Emergence and portrayal	Manipulation and control
Inductive and naturalistic	Deductive and experimental
Data analysis by themes from informants' descriptions	Statistical analysis
Data reported in language of informant	Statistical reporting
Descriptive write-up	Abstract impersonal write-up
Role of researcher	
Researcher as instrument	Researcher applies formal instruments
Personal involvement	Detachment
Emphatic understanding	Objective

3.2.2. Selected Approach

Qualitative research is appropriate where potential outcomes are not clear in advance (May, 2002) and where there is flexibility in the design or strategy of the researcher (Robson, 2002). For these reasons, and based on the strengths and weaknesses of the qualitative and quantitative methods outlined above, qualitative data collection and evaluation techniques were considered to be an appropriate approach for a part of this research – for example, in the development of the new concepts presented in this thesis. However, quantitative methods, also based on the strengths and weaknesses outlined in Table 6, are relevant to this research as they involve measurement and testing – directly applicable to cost estimation and customer satisfaction impact detection, and also in the replication of results. Therefore, it was necessary to utilise a combination of both qualitative and quantitative data collection and evaluation techniques in order to successfully proceed with the research project.

3.2.3. Data Collection

Surveys, literature review, case studies and interviewing are the main techniques used by qualitative researchers to collect data. All of these were used to gather the necessary information to complete this research project, along with ongoing and iterative experimentation via the development of a prototype software platform.

1. ***Literature Review*** - For qualitative researchers, the literature review is a stimulus for thinking, and not a way of summarising the previous work in the area that can blind the researcher to only considering existing concepts and conceptual schemes. New findings cannot always be fitted into existing categories and concepts, and the qualitative method encourages other ways of looking at the data. The literature review should be a sounding board for ideas, as well as for finding out what is already known and what specific methodologies have been used (Burns, 2000). The Literature Review is presented in Chapter 2 of this thesis.

2. ***Interviews*** - There are essentially three types of interview technique: structured, semi-structured, and unstructured.

   - Structured – these interviews have close-ended questions and are used predominantly in surveys and opinion polls. All respondents receive the same set of predetermined questions, asked in the same order so that comparisons can be made between the responses (Denzin and Lincoln, 1998). There is a lack of flexibility with this technique because the interviewee is obliged to select an answer from a limited set of responses previously established by the designer of the questionnaire. It can also destroy potential rapport between the interviewer and the interviewee, and leave no scope for discovering the beliefs, feelings or perceptions of the interviewee (Burns, 2000).

   - Semi-structured – these interviews have predetermined questions but the order, content, and wording can be modified depending on the interviewers’ perception of what appears appropriate. This allows greater flexibility than structured interviews and makes it easier to develop a rapport between interviewer and interviewee (Burns, 2000). The main problem with this technique is that it is difficult and time-consuming to assess comparisons of the information provided by the respondents.

- Unstructured - these interviews consist of open-ended questions and generally take the form of free-flowing conversation between interviewer and interviewee. The approach focuses on the informant's perception of themselves, of their environment, and of their experiences. Unstructured interviews rely heavily on the quality of the social interaction between the interviewer and interviewee, although this can be redirected by the interviewer if it should stray too far from the research study. Open-ended questions used with the unstructured approach increase flexibility, allow the researcher to go into greater depth or clear up any misunderstandings, encourage co-operation and rapport, allow the researcher to make a truer assessment of what the respondent really believes, and sometimes produce unexpected answers (Robson, 2002). The disadvantages lie in the possibility of loss of control by the interviewer, and in the difficulty of analysing the responses.

Table 7 provides a summary of the strengths and weaknesses of interviewing.

Table 7: Strengths and Weaknesses of Interviewing

Strengths	Weaknesses
Co-operation between interviewer and interviewee	Limited time period available with interviewee
Allows knowledge construction	Difficulties with establishing reliability
Provides in-depth understanding of informant's experiences	Time consuming to prepare questions and analyse data
Flexible/adaptable	Possible bias

3. ***Surveys*** - are a very common way of collecting data in which the participants answer a number of relevant questions. This is achieved in three main ways: a questionnaire is sent to the respondent by post or email and is completed (self-completion) by themselves; an interviewer delivers the questions in a face-to-face interview with the respondent and completes the questionnaire at the same time (face-to-face interview); or an interview is conducted via the telephone and the

interviewer records the responses (telephone interview). Robson states that surveys are not particularly suited to exploratory work, which may involve asking open-ended questions, but are best suited to completely standardised questions which mean exactly the same to every respondent – which can cause problems if the purpose is to explore an area (Robson, 2002). The following table, Table 8, briefly lists the disadvantages and advantages of the questionnaire-based survey:

Table 8: Disadvantages and Advantages of Surveys (Robson, 2002)

Disadvantages	**Advantages**
General to all surveys of this type	
Data affected by respondents' characteristics	A simple and straightforward approach to study of attitudes, beliefs, values, motives etc.
Respondents may not report beliefs, attitudes etc. accurately – trying to create good impression	Adaptable – can collect information from almost any human population
	High amounts of data standardisation
Postal and other self-administered surveys	
Low response rate. Sample may not be representative as don't know characteristics of non-respondents	Often only or easiest way
Ambiguities and misunderstandings in survey questions may not be detected	Can be very efficient – large amounts of data at low cost in short time
Survey may not be treated seriously – may not be able to detect this	Allow anonymity – can therefore encourage honest answers, particularly regarding sensitive areas
Interview surveys	
May be affected by interviewer's characteristics – could influence responses	Interviewer can ensure questions are clearly understood
May be affected by interactions of the interviewer and respondent	Presence of interviewer can encourage participation and involvement
Respondents may consider they are not anonymous and so be less honest and open	

Despite criticism, it is possible to use the survey as a part of an effective exploratory strategy – for instance, discovering how a number of organisations carry out a similar task by asking standardised questions, and gaining ideas or insights as a result of the similarities and differences in the answers. Also, in situations where the desired selection of survey respondents are not available for face-to-face or telephone interview due to time or distance limitations, the self-completion survey can work well as it allows the respondent to complete the questionnaire at their own convenience, and also allows them time to consider the questions and answers in more detail.

3.2.4. Research Evaluation

Unlike quantitative research which involves measurement and testing, assessment of qualitative research is a problem. One of the main problems concerns 'selective plausibilisation', in which the researcher uses only the parts of interviews or observation protocols that best illustrate the characteristics of the everyday world being studied. There is a lack of clarity in how the researcher handles cases and passages that he or she believes are not particularly illustrative of the characteristics, or cases and passages that might be contradictory. Therefore, the qualitative researcher needs to minimise this problem of selective plausibilisation, which can be achieved through the use of a variety of methods including expert opinion, case study, and reliability and validity (Flick, 2002).

1. ***Expert Opinion*** - The experts selected to evaluate the research should include people who have studied and worked in parts of the research area and who are capable of understanding the implications of the work carried out (Ince, 2000). The experts are employed to examine the research and then render a critical judgement on its outcome (Rossi and Freeman, 1993), as was the case in this project.

2. ***Case Study*** - The case study is relevant if there is a need to combine research with practice in the real world (Gill and Johnson, 2002) and is also recommended for all areas of information systems development and research, as in this instance (Howard et al, 2000). It is a useful technique when the research involves 'how' and 'why' questions, and can consist of a number of sources from which data can be collected: documentation, archival records, interviews, direct observation, participant observation, and physical artefacts (Yin, 2003).

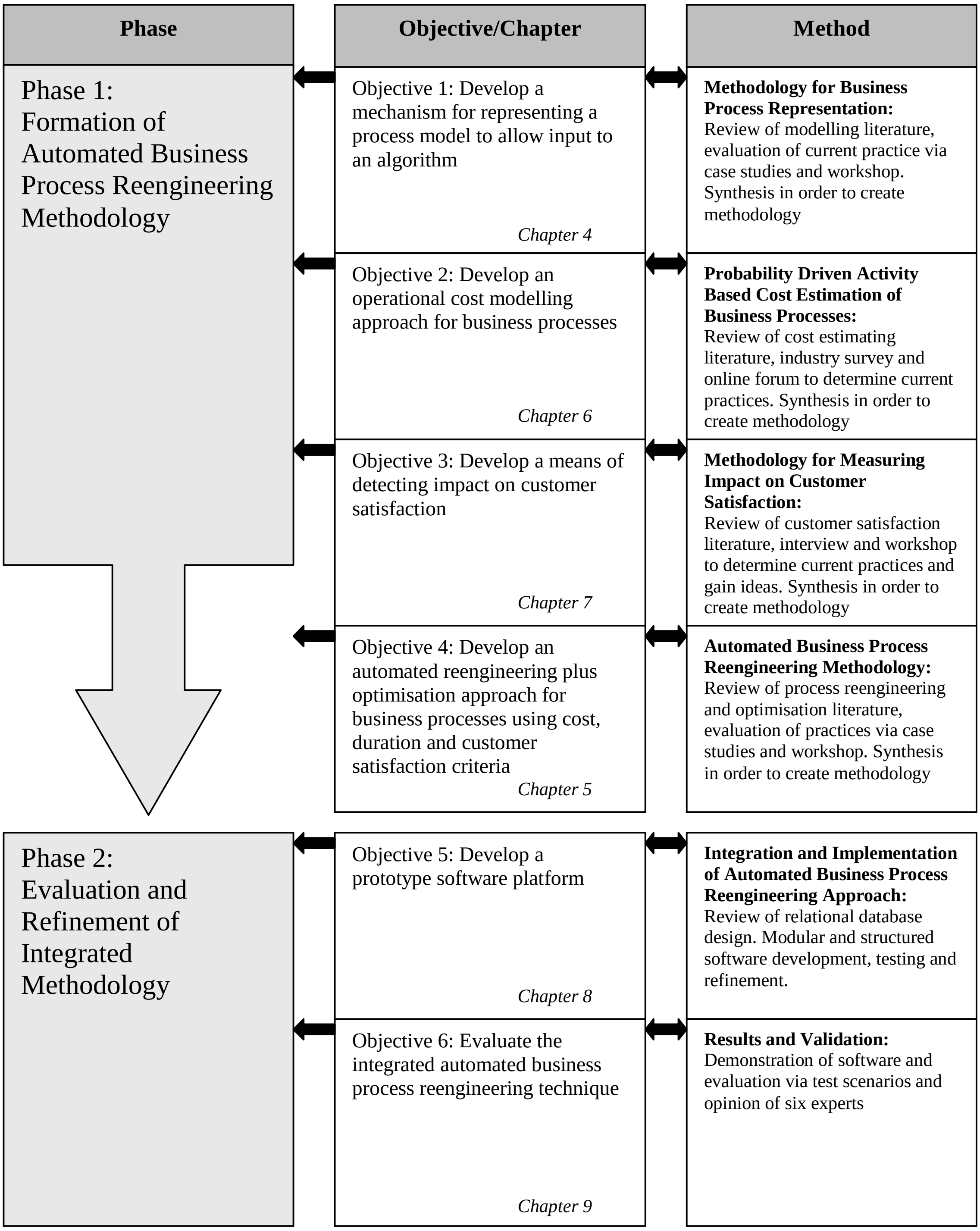
3. ***Reliability and Validity*** - Reliability can be defined as the extent to which a test or procedure produces similar results under constant conditions on every occasion

(Ince, 2000). Validity can be summarised as 'a question of whether the researcher sees what he or she thinks he or she sees' and that three errors may occur: to see a relationship where they are not correct; to reject them where they are correct; and to ask the wrong questions (Kirk and Miller, 1986).

4. ***Triangulation*** – This entails the use of multiple resources to help verify and validate qualitative research. By analysing data collected using different methods, the consistency of findings can be verified, and results can potentially be cross-checked (Jankowicz, 1995). Robson considers it to be a valuable and widely used method for increasing the validity of flexible research, but points out the possibility of discrepancies and disagreements between the different data sources (Robson, 2002).

3.3. Overall Methodology Adopted

As stated in Chapter 1, the purpose of this research is to develop a framework and technique that can be used by industry, namely the Service Sector. The research must also meet the requirements of the Doctor of Philosophy Programme – in that there should be a contribution to knowledge. A methodology that satisfies the demands of both industry and academia was selected. The selected methodology incorporates the techniques discussed in this chapter and broadly consisted of two principal phases, the formation of the automated business process reengineering methodology, and the evaluation and refinement of the integrated methodology. Figure 17 illustrates the two phases with the related research objectives, the methods used to achieve the objectives and the thesis chapters in which they are described.



Phase	Objective/Chapter	Method
Phase 1: Formation of Automated Business Process Reengineering Methodology	Objective 1: Develop a mechanism for representing a process model to allow input to an algorithm *Chapter 4*	**Methodology for Business Process Representation:** Review of modelling literature, evaluation of current practice via case studies and workshop. Synthesis in order to create methodology
	Objective 2: Develop an operational cost modelling approach for business processes *Chapter 6*	**Probability Driven Activity Based Cost Estimation of Business Processes:** Review of cost estimating literature, industry survey and online forum to determine current practices. Synthesis in order to create methodology
	Objective 3: Develop a means of detecting impact on customer satisfaction *Chapter 7*	**Methodology for Measuring Impact on Customer Satisfaction:** Review of customer satisfaction literature, interview and workshop to determine current practices and gain ideas. Synthesis in order to create methodology
	Objective 4: Develop an automated reengineering plus optimisation approach for business processes using cost, duration and customer satisfaction criteria *Chapter 5*	**Automated Business Process Reengineering Methodology:** Review of process reengineering and optimisation literature, evaluation of practices via case studies and workshop. Synthesis in order to create methodology
Phase 2: Evaluation and Refinement of Integrated Methodology	Objective 5: Develop a prototype software platform *Chapter 8*	**Integration and Implementation of Automated Business Process Reengineering Approach:** Review of relational database design. Modular and structured software development, testing and refinement.
	Objective 6: Evaluate the integrated automated business process reengineering technique *Chapter 9*	**Results and Validation:** Demonstration of software and evaluation via test scenarios and opinion of six experts

Figure 17: Research Programme

The main elements of data collection and evaluation in this research, as reviewed in Sections 3.2.3 and 3.2.4, briefly included the following:

1. ***Literature Review*** – this was conducted initially to provide a foundation for the research to follow and to generate ideas. It was then conducted on an ongoing basis throughout the entire project (Chapter 2).

2. ***Interviews*** - following an assessment of the interview techniques summarised, together with their overall strengths and weaknesses, a combination of semi-structured and unstructured interviewing was considered to be most appropriate for this research. It was not advisable to limit any information available in the subject area which interviewing was applied to, the measurement of customer satisfaction, since the desire was to achieve a broad overview of the area, despite the potential problems with these techniques, as briefly outlined in Section 3.2.3. (Chapter 7).

3. ***Surveys*** - The self-completion survey technique was employed for a part of the data collection in this project due to the reasons mentioned in Section 3.2.3 – mainly, availability of respondents and the potential for more detailed answers via this method. The survey formed a basis for the cost estimating methodology to be developed. An online forum was utilised as a secondary, more specific survey (Chapter 6).

4. ***Expert Opinion*** – Experts were identified from industry with expertise in all or some of the relevant areas of research. They were utilised for validation of the research (Chapter 9).

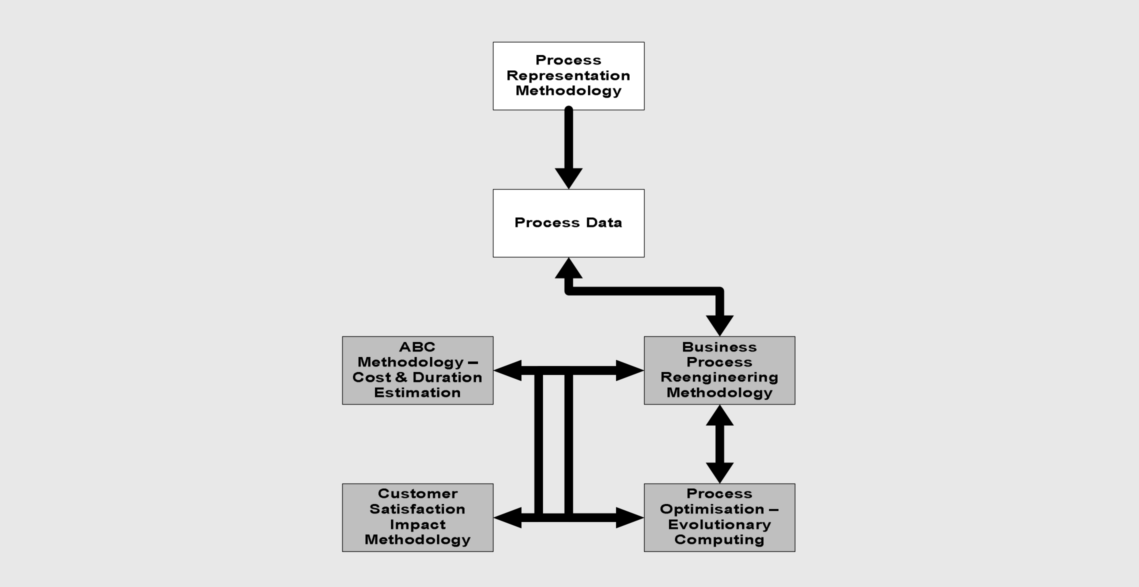
5. ***Case Study*** **–** two case studies were conducted with the industrial sponsor to identify current research and development work plus technology and software tools being used. Documentation and software were collected, followed by a period of familiarisation and experimentation. A workshop was also conducted in between the two case studies (Chapter 4).

3.4. Summary

This chapter detailed the development of the three-phase research methodology used to achieve the aim and objectives of the project, which were derived from the research gaps identified in Chapter 2. It reviewed quantitative and qualitative research methods, the research methodology utilised, and the data collection and evaluation methods adopted in order to complete the research.

The next chapter begins with an overview of current research and development work at the industrial sponsors, followed by details of case studies and a workshop conducted. This leads to the presentation of the individual elements of the overall reengineering technique, starting in Chapter 4: the methodology for business process representation (Chapter 4); the automated reengineering methodology (Chapter 5); the methodology for cost estimation of business processes (Chapter 6); and the customer satisfaction impact methodology (Chapter 7). Chapter 8 presents their integration and implementation.

4. Methodology for Business Process Representation


Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

This chapter begins by providing a background to the development of the business process representation methodology (to be described later in the chapter), and the methodologies presented in Chapters 5, 6 and 7. Firstly, an overview of current research and development by the industrial sponsors is presented, followed by details of a workshop, along with the case studies which led to the acquisition of a selection of typical service sector process models.

4.1. Current Research and Development at British Telecom

The main areas of current research and development at BT involve business process reengineering, optimisation and mining, churn prediction and contact centre environment improvement. Soft computing techniques, such as evolutionary computing, neural networks and fuzzy logic, are being employed to assist in this research and development work.

One particular tool that provides a basis for much research into process optimisation and mining at BT is jBPM, a BPMS developed by JBoss, previously mentioned in Chapter 2. It allows the user to model a business process (in Eclipse) using a series of nodes, states, tasks, actions, transitions and decisions, and also to manage the execution of the process. All the elements of a jBPM process model can be stored in a relational database (PostgreSQL), plus details of the process execution, or instance. This information can then be retrieved and analysed. Figure 18 shows a jBPM process model.

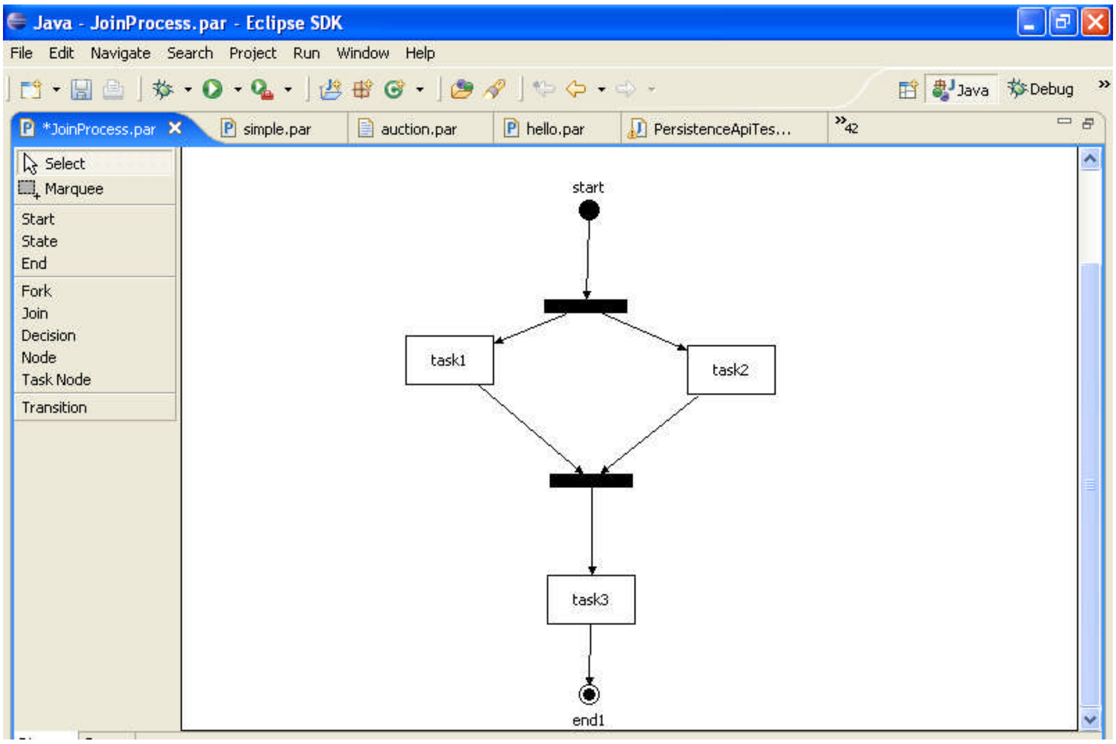


Figure 18: Current R & D at British Telecom – JBoss jBPM

A 'drag and drop' facility allows the creation of process models which can also include branches into parallel or alternate flows. The actual model is held as XML code. As jBPM is an open source tool, BT is able to customise and manipulate it, using JAVA, as a part of their research and development work into process optimisation and mining. However, a problem with this current situation is the retrieval, restructuring and analysis of process data – thus the reason for research into workflow mining currently being undertaken.

4.2. British Telecom – Case Studies & Workshop

Following initial literature review, one week was spent with BT during which time a case study was conducted. As overviewed in Section 4.1., the author was introduced to BT's current practices, jBPM was demonstrated and supplied to the author for familiarisation purposes, process data was provided, and information was gathered on business processes in the service sector. At the end of this time with the sponsor, a workshop was held with two senior BT research and development managers (the industrial supervisors for this research), four BT research and development officers, plus a doctoral researcher and the two academic supervisors of this research from Cranfield University. The purpose of the workshop was to expand on the initial research brief and to determine the finite details of the research to follow. The main findings from the first case study and workshop were as follows:

- The service sector is process driven
- Organisations such as BT have very large numbers of processes
- Many of these processes are automated (i.e. web-based)
- These processes involve large numbers of people
- Customer satisfaction is a key factor in the service sector
- The time involved in reengineering of processes should be as short as possible to respond to customer and market demand

Based on some of these findings and a great deal of discussion, the main conclusions from the workshop were:

- Extend the automated BPR to include an optimisation facility in order to increase its effectiveness (Chapter 5).
- Establish process cost reduction as the primary objective of the research to follow (Chapter 6).
- Make customer satisfaction (Chapter 7) and process duration (Chapter 6) constraints on the cost reduction – for instance, a cheaper process is not necessarily a better one

from a customer perspective. Any negative impact on customer satisfaction should involve rejection of a reengineered process, even if its cost is reduced.

Therefore, this led to the identification of the process related objective and constraints of the work to follow – notably that, although three values are involved, this was to be a single objective reengineering framework. The generation of these three values is presented in Chapters 6 and 7, and how they link together is presented in Chapter 8. The workshop also provided suggestions concerning the achievement of the overall aim of the research, particularly via the possibility of customising jBPM to achieve this, or at least taking ideas from it. The workshop also enabled re-validation of the research questions of the project. The first case study was later followed by a second one in which more process data was collected, eventually leading to the development of a process library, presented in Section 4.5, and ideas concerning representation of the process models in the library in order to allow their input to a reengineering algorithm. The methodology developed as a result is now described.

4.3. Business Process Representation

In the early stages of the research, a number of criteria were set by the author for an ideal process representation technique. This was necessary due to the large number of process modelling tools and techniques available, plus their wide variety of features, in order to target the review more effectively. They were as follows:

- The process representation should be readable and understandable without specialised knowledge, such as computer science or mathematics.

- It should be possible for a user without specialised knowledge to amend or create process models (with the assistance of a GUI).

- It should allow quantitative (measurable) analysis rather than purely qualitative analysis – it is not possible to examine and optimise a process via algorithmic means without quantitative information (for example, the duration of a task). Therefore, it should be represented with an appropriate level of detail to allow this.

- It should allow representation of complex processes - service sector business processes are based on human to human or human to machine (i.e. web-based processes) interaction and, as a result, may involve branches and/or repetitions based on decisions and events.

- The model should contain sufficient information to analyse each element of the process, individually or in sections, and compare with other process data held in a data warehouse, in order to potentially reengineer to the extent where complete process reconfiguration could occur.

- The detailed (dynamic and functional), quantitative process model should allow storage in a database and retrieval for purposes of analysis (it should be possible to store the model and then retrieve it in exactly the same state as it was stored in).

- It should be possible to link cost and duration measures to the steps in the model

However, none of the modelling tools and techniques reviewed caters for all of the criteria, as discussed in Chapter 2. It was therefore necessary to develop a technique which incorporated all of the desired features. The proposed business process representation structure and methodology attempts to address the research gap identified in Section 2.10: a requirement for the development of a quantitative complex process representation mechanism which enables input, analysis and evaluation by an algorithm, and which also contains sufficient information to allow comparisons to be made with all available process data to ascertain its suitability for replacements of parts of existing processes

4.4. Business Process Representation Methodology

Ideas were generated by the review of process modelling techniques, notably the flow diagram due to its ability to depict process flow at the desired level of detail, along with WFMS and the facility to transform diagrammatic representations into computer code in order to enable execution of the process. Further research into group technology, classification and coding systems, and taxonomies led to the creation of the representation methodology. This differs from other representation techniques in that it is a purely numeric representation – it is not diagrammatical or mathematical, it is simply a string of codes, one for each element of the process, which, when strung together, creates a comprehensive numerical process model containing details of all inputs, outputs, constraints, resources used, type of element and flows in and out, next/previous steps, and a unique identifier for each element based on the creation of a company taxonomy. This methodology caters for all of the criteria listed above and is described in the following sections, 4.4.1 to 4.4.3.

4.4.1. Taxonomy

This research requires a specific, multi level taxonomy in order to allow the assignment of codes to the actual process and its component steps at the bottom of the hierarchy. This entails an extensive breakdown of a business structure from the highest level down to the lowest level. A case study of a service based company (the industrial sponsor) was conducted and a company taxonomy was designed, based on the data gathered during the study. The complete company taxonomy is very large and so only parts of the Customer Service route are used, as is appropriate for this research, to illustrate certain example processes. The example in Figure 19 shows one section of the taxonomy, detailing the Customer Service/Contact/Orders/Modify Order levels of the structure.

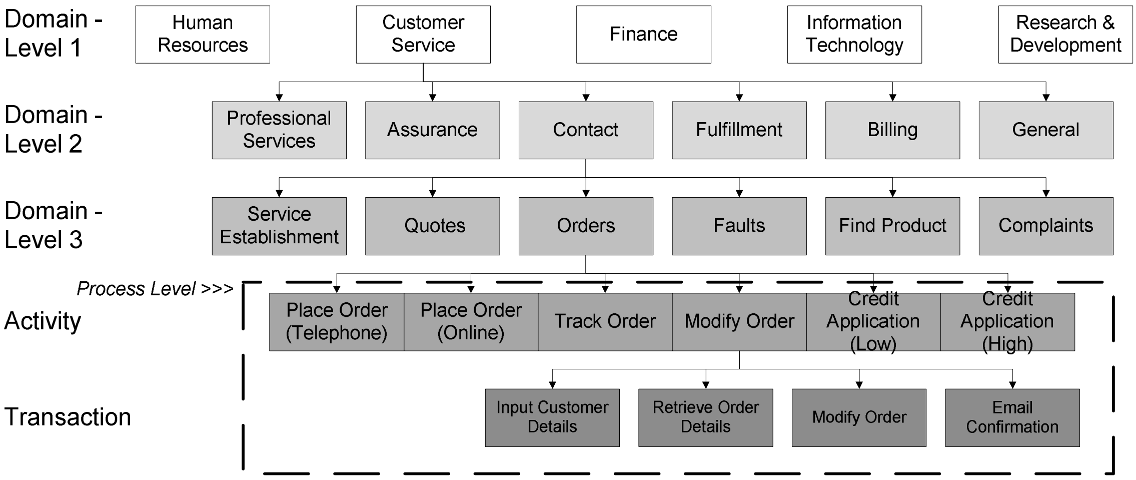


Figure 19: Example Service Sector Taxonomy

In this example, 'Customer Service' is a root node (although the company itself would be the root node, but this does not require specifying in a company taxonomy) and a domain, along with 'Human Resources', 'Finance', 'Information Technology' and 'Research & Development'; 'Customer Service' leads to sub-domains which include 'Contact'; this branches to a third level of domains, 'Service Establishment', 'Quotes', 'Orders', 'Faults', 'Find Product' and 'Complaints'; 'Orders' branches to six possible activities which can then be broken down into transactions, as illustrated with activity 'Modify Order', which is used as one of the example process models.

It is necessary to break down a taxonomy such as this into one more level of detail ('Sub-Transaction' or 'Transaction – Level 2') in order to fully represent a process - this is explained in Section 4.4.3.

There may be a degree of overlap amongst the categories in taxonomies like this one, where activities and/or transactions are shared between different domain levels. As stated by Sayal, a process instance can only belong to one category within a taxonomy at any time (Sayal et al, 2002), but this applies to the actual process execution (the instance) and not the representation – there is no reason why elements of a process cannot appear in numerous models. For instance, the process step, 'Email Confirmation', may well appear in a large number of the processes within the company taxonomy. These overlaps are taken into account in the intended functionality and the search space extends into all the different areas of the taxonomy.

4.4.2. Taxonomy Coding Mechanism

Descriptive codes must be assigned to each level and each element of the taxonomy. Only the parts of the taxonomy shown in Figure 19 are coded for this example. Table 9 shows a simple, purely numeric taxonomy coding system which is suitable for the aims of the project.

Table 9: Taxonomy Coding System

CODE	LEVEL	NAME
1	DOMAIN L1	HUMAN RESOURCES
2	DOMAIN L1	CUSTOMER SERVICE
3	DOMAIN L1	FINANCE
4	DOMAIN L1	INFORMATION TECHNOLOGY
5	DOMAIN L1	RESEARCH & DEVELOPMENT
1	DOMAIN L2	PROFESSIONAL SERVICES
2	DOMAIN L2	ASSURANCE
3	DOMAIN L2	CONTACT
4	DOMAIN L2	FULFILMENT
5	DOMAIN L2	BILLING
6	DOMAIN L2	GENERAL
01	DOMAIN L3	SERVICE ESTABLISHMENT
02	DOMAIN L3	QUOTES
03	DOMAIN L3	ORDERS
04	DOMAIN L3	FAULTS
05	DOMAIN L3	FIND PRODUCT
06	DOMAIN L3	COMPLAINTS
01	ACTIVITY	PLACE ORDER (TELEPHONE)
02	ACTIVITY	PLACE ORDER (AUTOMATED/ONLINE)
03	ACTIVITY	TRACK ORDER
04	ACTIVITY	MODIFY ORDER
05	ACTIVITY	CREDIT APPLICATION (LOW)
06	ACTIVITY	CREDIT APPLICATION (HIGH)
01	TRANSACTION	INPUT CUSTOMER DETAILS
02	TRANSACTION	RETRIEVE ORDER DETAILS
03	TRANSACTION	MODIFY ORDER
04	TRANSACTION	EMAIL CONFIRMATION

In this example, the code numbers are all sequential and start from 1. This may not be as straight forward had the entire company taxonomy been coded. The highest levels of the taxonomy have the least categories, thus the allowance for only one digit in the first two domain levels; this is increased to two digits in the lower levels.

4.4.3. Process Coding Mechanism

Now that the taxonomy has been assigned the necessary codes to identify its component domains, activities and transactions, it is necessary to construct a process flow diagram from the codes used. The taxonomy representation of 'Modify Order' is as follows in Figure 20:

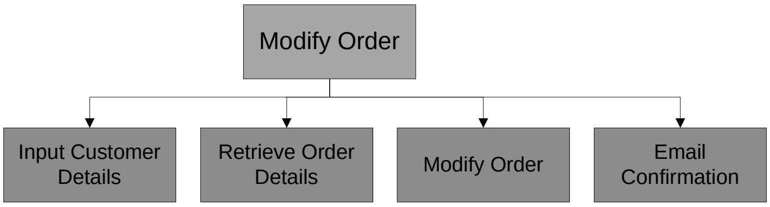


Figure 20: Activity 'Modify Order'

A process may contain one to many activities – in the examples presented, the activity is the only one and so is referred to as a process. The steps of the process have already been defined to a large degree in the taxonomy, but it is not possible to depict such elements as loops, decisions, branches etc. in the taxonomy structure - for instance, the transactions that constitute the above process need to include many of these elements. The actual process flow diagram, modelled using OMG's BPMN (see Chapter 2 – Section 2.1. and 2.9.1.), is more complex and is shown in Figure 21 below.

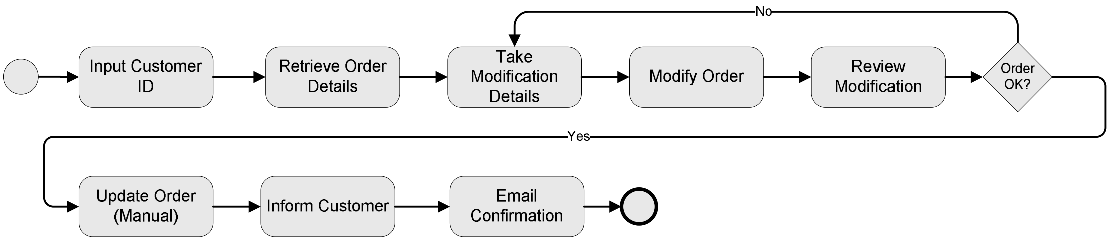


Figure 21: Modify Order Process Model

Extra considerations are required to fully represent the complete process. Although the order of the process and its steps are exactly as in the taxonomy definition, another level of taxonomy is needed to break the transaction level down into more detail, such as checking the modification details. As such, these extra elements in the process also require codes so they can be included in the numerical process representation. This lowest level of the taxonomy is classed as the sub-transaction level. All sub-transactions are referred to as process ‘steps’ unless it is necessary to be more specific. Therefore, the process representation encapsulates all activities, transactions, sub-transactions and flows.

The numerical representation of each step of a process is listed below in Table 10 with some examples and definitions. This mechanism requires a total of 174 digits per step.

Table 10: Process Code Details

CODE NAME	NO. DIGITS	EXAMPLE/EXPLANATION
Domain Level 1	1	1=Human Resources
Domain Level 2	1	5=Billing
Domain Level 3	2	04=Faults
Activity	2	03=Track Order
Transaction	2	03=Order Product
Sub-Transaction	2	03=Check Postal Code
Input	15	Up to 3 possible alternative inputs of 5 digits E.G. 002050020600000=Placed Order or Placed Delivery
Output	15	Up to 3 possible alternative outputs of 5 digits E.G. 002030020400000=Pending Order or Pending Delivery
Constraint	9	Up to 3 constraints of 3 digits E.G. 001003000=Must be first step AND must be linked to next step
Resource	3	001=Internal Customer RDB
Flows In	10	Up to 5 incoming flows of 2 digits per flow (normally 1 except for join (AND-Join) & merging (OR-Join) process steps). E.G. 0101000000=2 incoming normal flows
Flows Out	10	Up to 5 outgoing flows of 2 digits per flow (normally 1 except for fork (AND-Split) & decision (OR-Split) process steps) E.G. 0109010000=1 outgoing normal flow, 1 sequence flow loop, 1 outgoing normal flow
Previous Step	50	Allows for up to 5 steps of 10 digits (Domain L1, Domain L2, Domain L3, Activity, Transaction, Sub-Transaction)
Next Step	50	Allows for up to 5 steps of 10 digits (Domain L1, Domain L2, Domain L3, Activity, Transaction, Sub-Transaction)
Step Type	2	20=Gateway – AND-Join

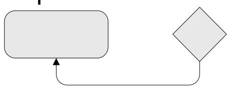
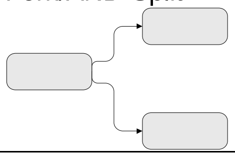
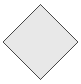
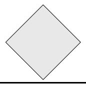
The codes Input, Output, Constraint, Resource, Flows In, Flows Out and Step Type are crucial to the eventual evaluation of each element of the process. They have a range of possible values, particularly input and output codes which are derived from resource centres – resource centres are a part of the cost estimating structure and are explained in

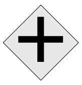
Chapter 6. These are some examples of the codes:

• Constraint	001=First Step
	002=Last Step
	003=Must be Attached to Next
	004=Must be Attached to Previous
	005=Must be Present
	006=Duplicate OK
• Resource Centre	01=Internal DB Systems
	03=Communications
	04=Shipping
• Input Code	01001=Customer Surname
	01002=Customer ID
• Output Code	01005=Customer Account

With the exception of Next Step, Previous Step and Flows In/Out, the codes that constitute the 174 digit process step code are fixed – they do not change during the execution of the reengineering technique. Also, it should be noted that the 3 possible Input and Output codes allowed for are alternatives - only one is used as an input and only one produced as an output, but this allows for multiple inputs and outputs that are similar in nature to be included and considered during execution of the technique. Flows In/Out and Step Type code definitions are taken from OMG's BPMN terminologies. The available code values and their definitions, plus example BPMN modelling notations (illustrated where used in later examples) are as follows in Table 11:

Table 11: BPMN Flow and Step Types

NAME	VALUE	DEFINITION
Flows In/ Out	01	Normal Flow
	02	Uncontrolled Flow
	03	Conditional Flow
	04	Default Flow
	05	Exception Flow
	06	Message Flow
	07	Compensation Association
	08	Activity Loop
	09	Sequence Flow Loop 
	10	Fork/AND-Split 
Step Type	01	Start Event
	02	Intermediate Event
	03	End Event
	04	Message trigger
	05	Timer trigger
	06	Error trigger
	07	Cancel trigger
	08	Compensation trigger
	09	Rule trigger
	10	Link trigger
	11	Multiple trigger
	12	Terminate trigger
	13	Task
	14	Collapsed Sub-Process
	15	Expanded Sub-Process
	16	Gateway: Exclusive (XOR) Data-Based - Decision/Merging 
	17	Gateway: Exclusive (XOR) Event-Based - Decision/Merging
	18	Gateway: Inclusive (OR) - Decision/Merging
	19	Gateway: Complex
	20	Gateway: Parallel (AND) – Forking 

	21	Gateway: Parallel (AND) - Joining 
	22	Data Object
	23	Multiple Instances
	24	Process Break
	25	Transaction
	26	Nested/Embedded Sub-Process
	27	Group
	28	Association
	29	Pool
	30	Lane

As an example, the task step ‘Review Modification’ from the process, ‘Modify Order’, shown in Figure 21, is constructed as follows:

• Domain Level 1	2	(Customer Service)
• Domain Level 2	3	(Contact)
• Domain Level 3	03	(Orders)
• Activity	04	(Modify Order)
• Transaction	03	(Modify Order)
• Sub-Transaction	03	(Review Modification)
• Input (1)	01018	(Modified Order – Unchecked)
• Input (2)	00000	
• Input (3)	00000	
• Output (1)	01019	(Modified Order – Checked)
• Output (2)	00000	
• Output (3)	00000	
• Constraint (1)	000	
• Constraint (2)	000	
• Constraint (3)	000	
• Resource	004	(Internal Orders RDB)
• Flows In (1)	01	(Normal Flow)
• Flows In (2)	00	
• Flows In (3)	00	
• Flows In (4)	00	
• Flows In (5)	00	
• Flows Out (1)	01	(Normal Flow)
• Flows Out (2)	00	
• Flows Out (3)	00	
• Flows Out (4)		00
• Flows Out (5)	00	
• Previous Step (1)	2303040302	(Modify Order)
• Previous Step (2)	0000000000	
• Previous Step (3)	0000000000	
• Previous Step (4)	0000000000	
• Previous Step (5)	0000000000	
• Next Step (1)	2303040304	(Order OK?)
• Next Step (2)	0000000000	
• Next Step (3)	0000000000	

Next Step (4)	0000000000	
Next Step (5)	0000000000	
Step Type	13	(Task)

The codes are then strung together and the task step is represented, as in Figure 22 below:

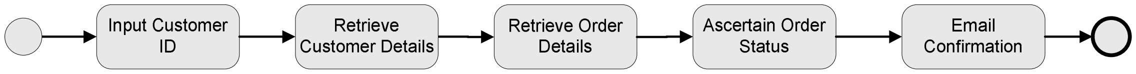
```
23030403030021800000000000021900000000000000000000040100000000001000000023030403020000000000
0000000000000000000000000000002303040304000000000000000000000000000000000000000013
```

Figure 22: Modify Order - Task Code

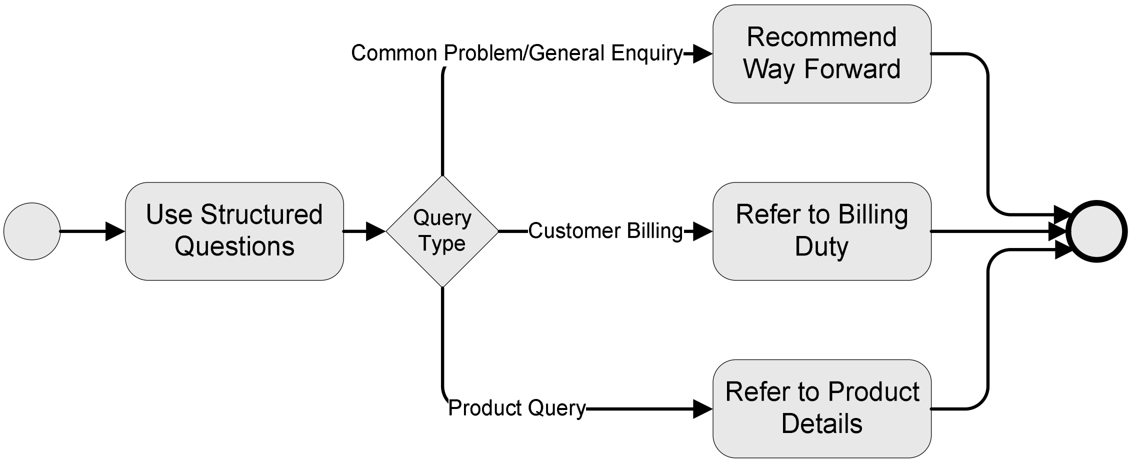
The entire process translates into a code of 1914 digits – 11 process steps each with a 174 digit code as above. This string of digits can than be input to the automated reengineering algorithm where its related cost and duration data can be retrieved and the whole process can be analysed and evaluated with a view to reengineering (Stelling et al, 2008a). This is described in the following chapter.

4.5. Process Library

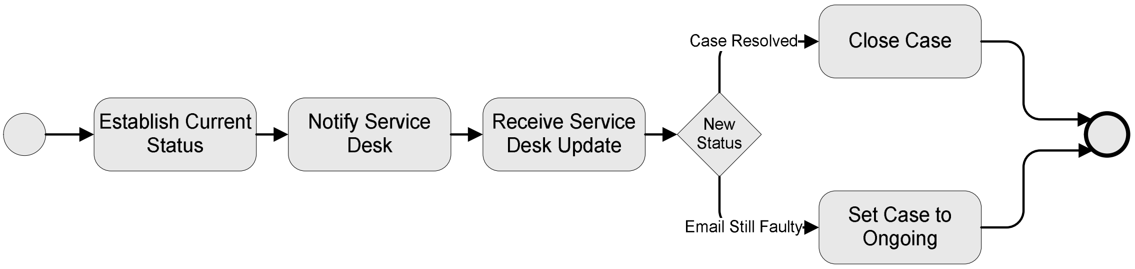
Using the business process representation methodology, a library of processes of varying complexity was developed for testing and demonstration purposes, along with a multitude of process steps and sub-processes which may provide replacements for parts of existing processes in the library – the way they are used is described in the following chapter. All process data is held in a relational database, which is described in detail in Chapter 8. The process library, modelled using OMG's BPMN, is based on process data provided by the industrial sponsor and represents a typical sample of service based processes. Generic processes and sub-processes were also developed for testing and demonstration purposes, many of which are shown in Chapters 5, 6 and 7. The library comprises of the following processes (Figure 22):



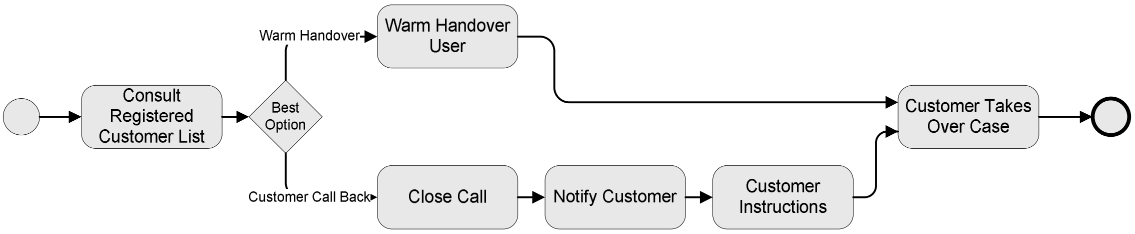
Track Order



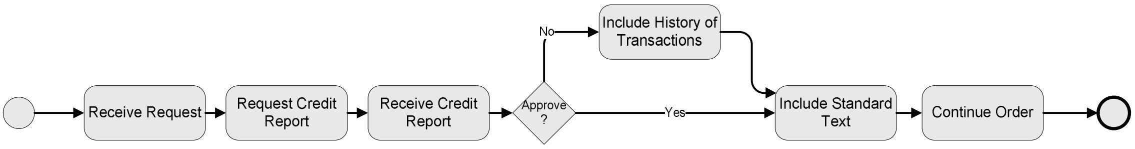
Customer Advice Request



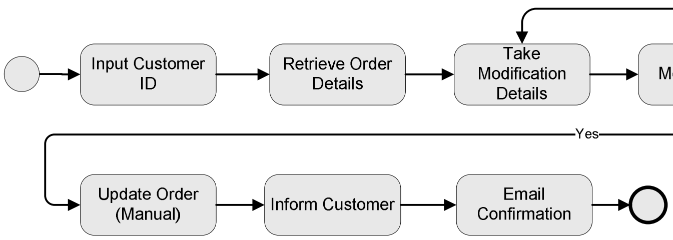
Resolve Platform Problem



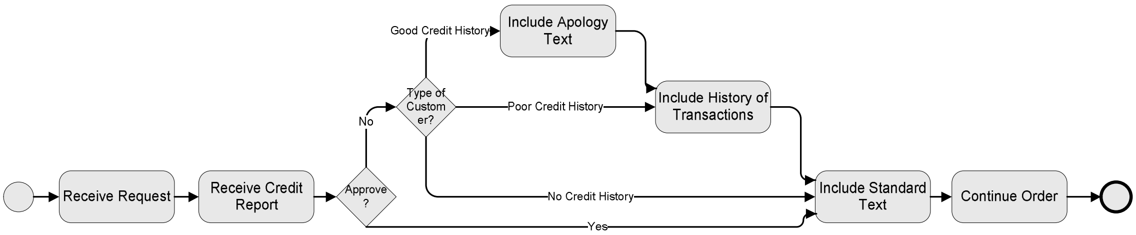
Pass to Customer



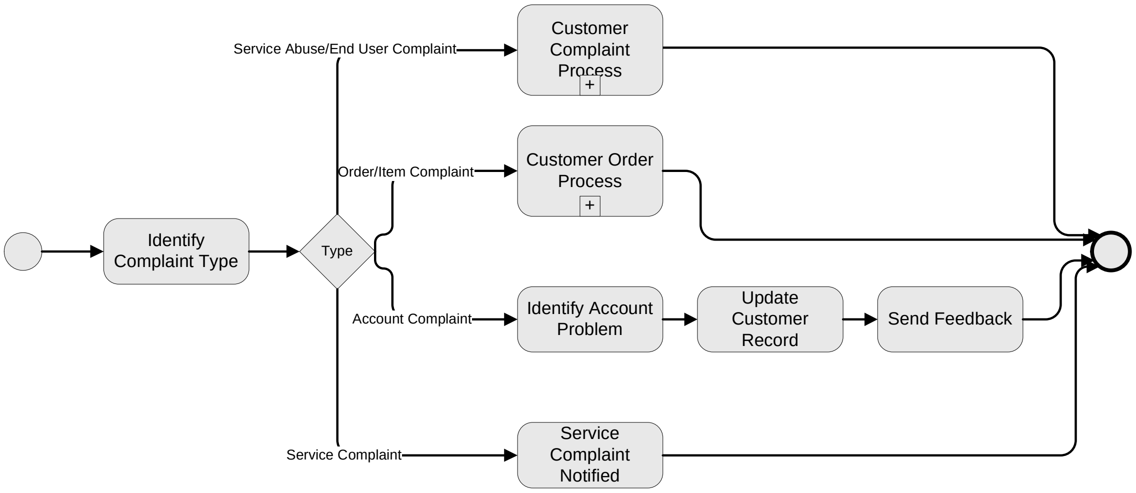
Credit Application – Level 1



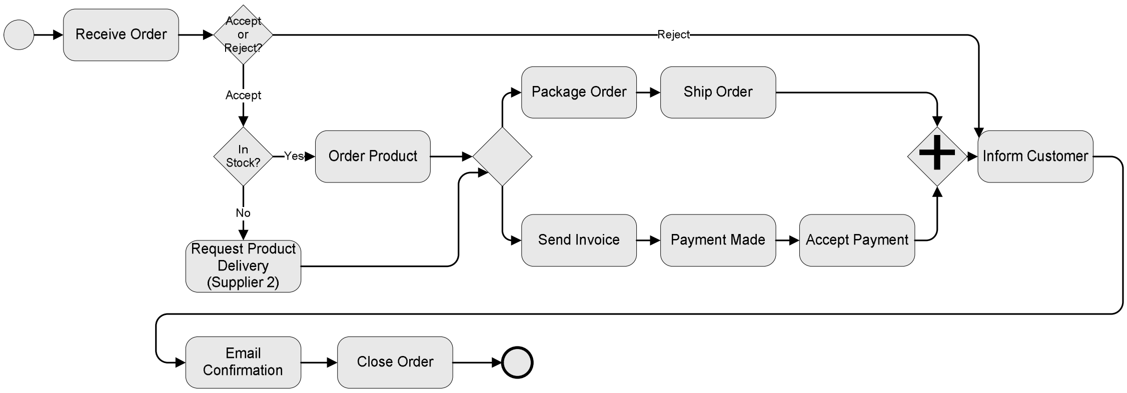
Modify Order



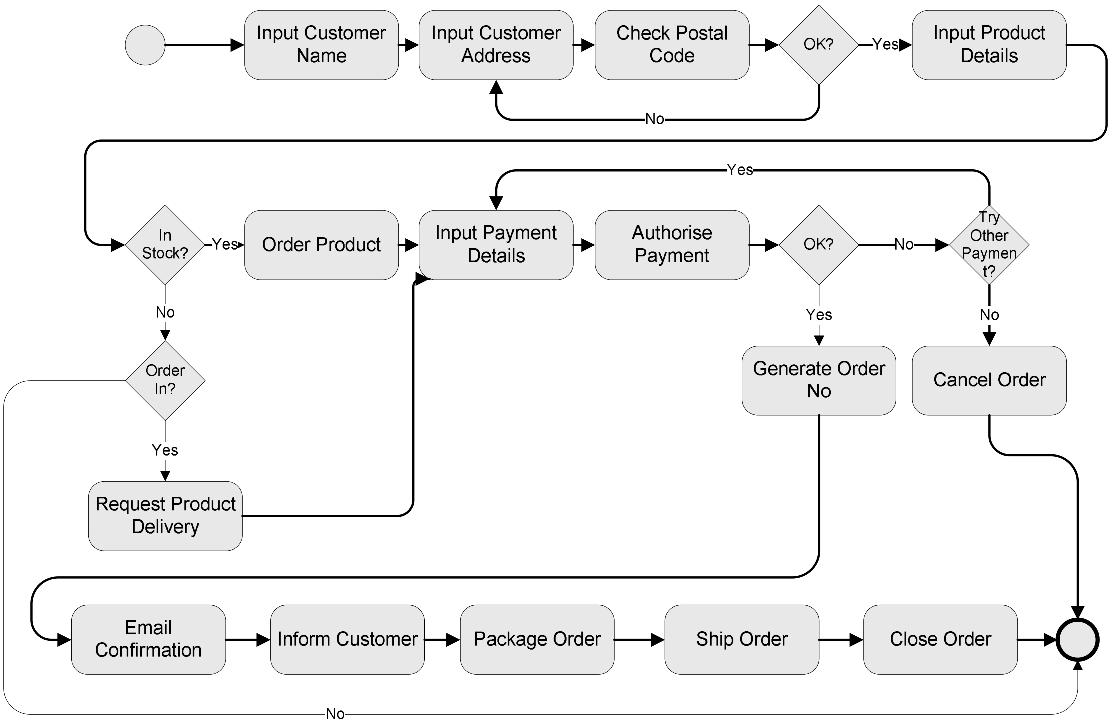
Credit Application – Level 2



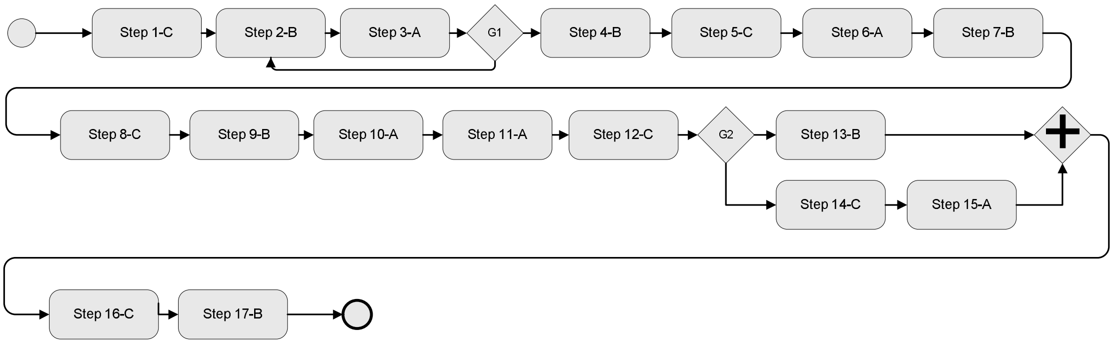
Private Customer Complaint



Automated/Online Order



Place Telephone Order



Generic Process

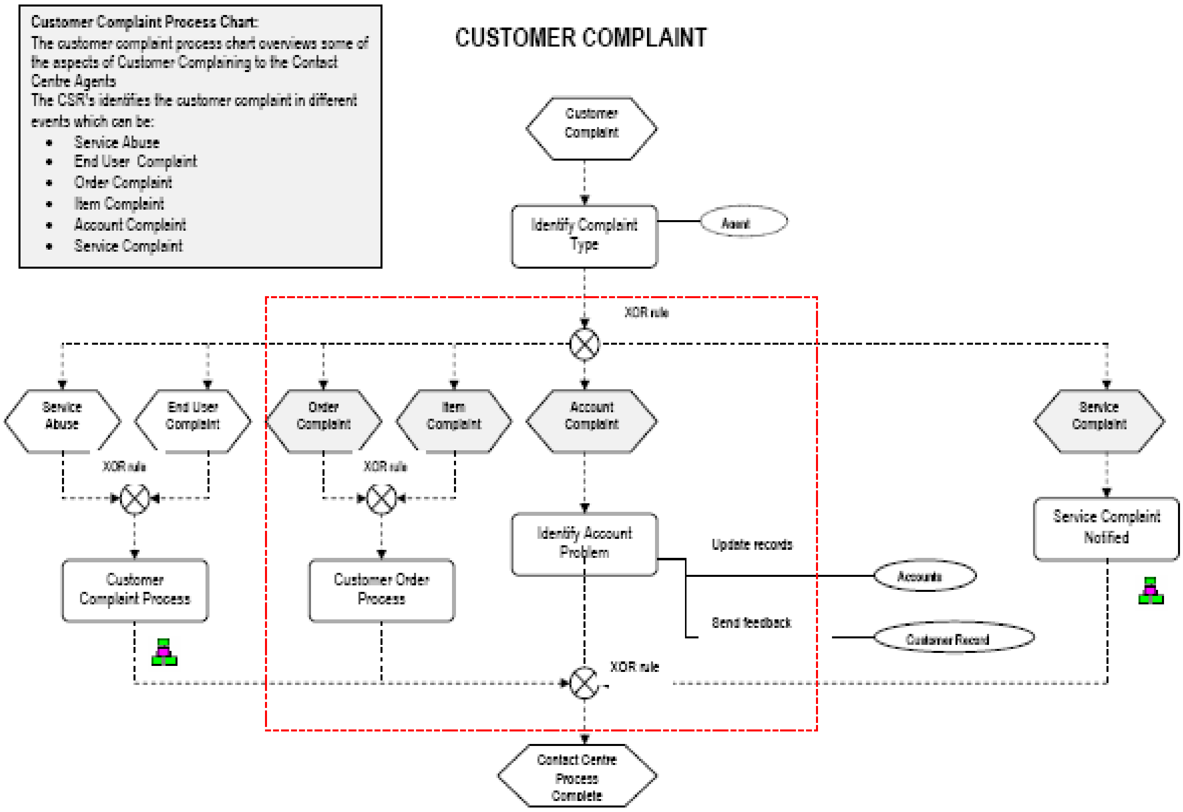
Figure 23: Process Library

Some of the above processes are used to demonstrate the automated reengineering technique and its component methodologies, and also are presented in Chapter 9 as test scenarios. All process models are stored in a relational database, presented in Chapter 8. Some of the models in the process library were translated from ARIS – the next section gives an example of how an ARIS and an IDEF3 can effectively be translated into the BPMN flow diagrams such as the ones shown in Figure 23.

4.6. Translation of Process Models from ARIS and IDEF3

As a process model should depict a logical sequence of events, it should therefore be possible to translate many of the different representations into BPMN flow diagrams. This translation may be necessary in order to facilitate the implementation of the process representation methodology described in this chapter. This was successfully attempted on an ARIS (Figure 24) and an IDEF3 (Figure 25) diagram (both previously shown in Chapter 2), as follows:

ARIS – CUSTOMER COMPLAINT PROCESS MODEL:



Customer Complaint Process Chart:
The customer complaint process chart overviews some of the aspects of Customer Complaining to the Contact Centre Agents
The CSR's identifies the customer complaint in different events which can be:

- Service Abuse
- End User Complaint
- Order Complaint
- Item Complaint
- Account Complaint
- Service Complaint

BPMN FLOW DIAGRAM:

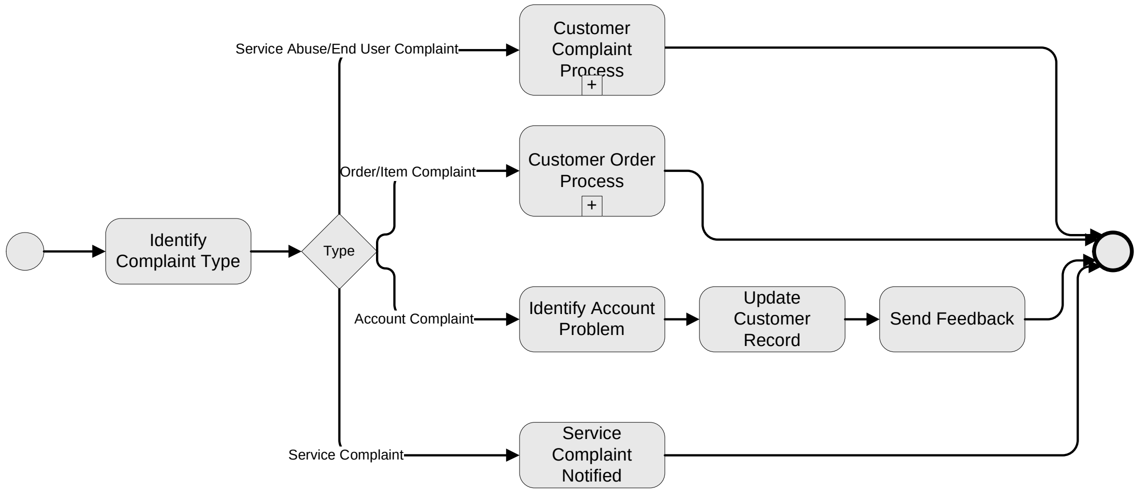
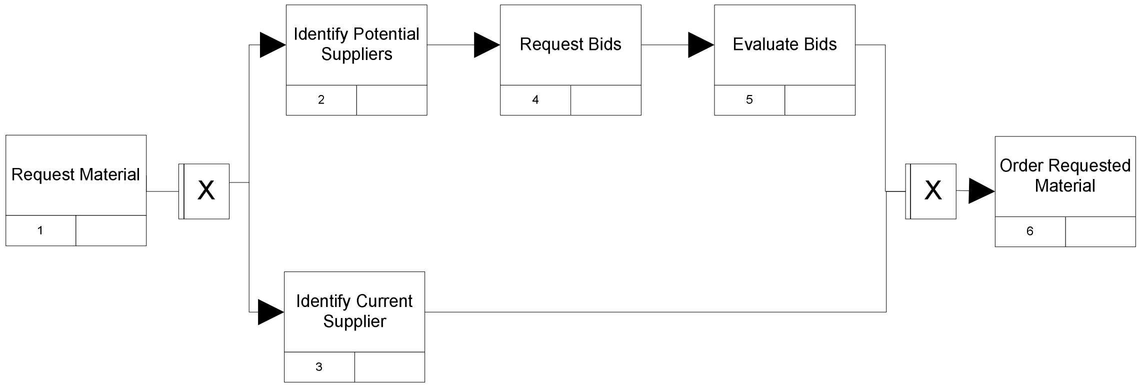


Figure 24: ARIS Translation to BPMN Flow Diagram

IDEF3 – MATERIAL ORDER PROCESS MODEL:



BPMN FLOW DIAGRAM:

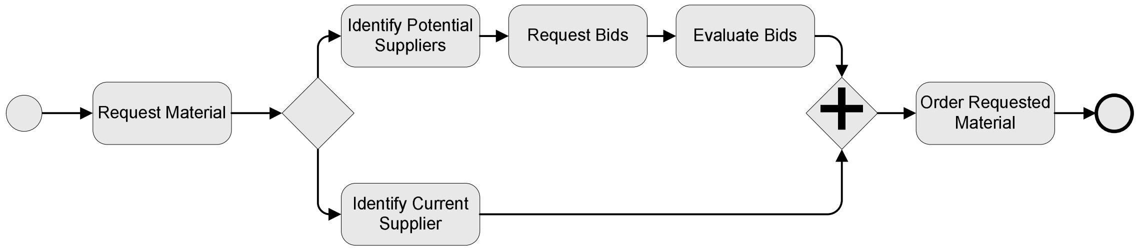


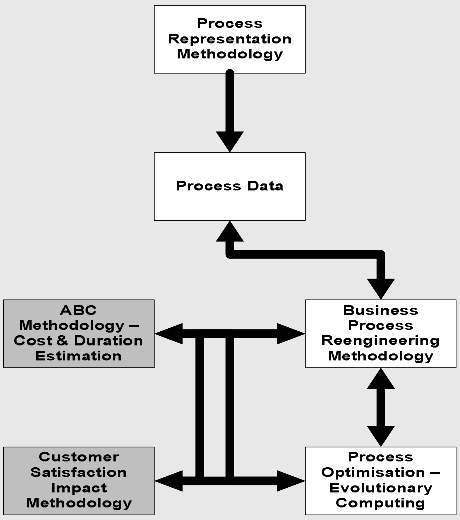
Figure 25: IDEF3 Translation to BPMN Flow Diagram

4.7. Summary

This chapter described the background to the development work in this research, and presented the business process structure and representation methodology. The methodology can feasibly be used independently of the proposed implementation structure. An organisation’s existing process models can be translated into BPMN flow diagrams and then coded, or directly coded from the original model, and each step assigned a unique 10 digit identifier without the need to undergo the complete taxonomy related implementation. As all process models, whatever format, should depict a logical sequence of events, it should be possible to translate them all into a BPMN flow diagram and then into a coded representation, or alternatively, directly into a coded representation. Depending on the format, this involves varying degrees of difficulty. It has been verified by translating IDEF3 and ARIS diagrams into BPMN diagrams, as demonstrated in Section 4.6. Once translated, the coding procedure is then simpler than a direct translation might be.

The next chapter describes how this coded representation is dealt with by the automated reengineering methodology.

5. Automated Business Process Reengineering Methodology


Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

This chapter describes the methodology which uses the coded process data presented in Chapter 4 as input. Literature review enabled the identification of the levels, rules and logic of BPR; case studies and workshop, as presented in the previous chapter, created an understanding of the issues relating to service based processes; and the combination of this knowledge led to the development of a step-by-step BPR procedure, aimed at a process model level. Initially, the basic logic for this BPR procedure was implemented via the development of an algorithm, later followed by the inclusion of an optimisation module – this combination not only automated the BPR rules and logic, but also increased the efficiency by allowing the exploration of considerably more search space, via the inclusion of a GA, than could feasibly be achieved manually. During qualitative validation, experts agreed unanimously that the technique to be presented in this chapter is undoubtedly more efficient than a manual attempt.

The proposed automated BPR methodology attempts to address the research gap identified in Section 2.10: a requirement for the development of a complex process reengineering, optimisation and reconfiguration technique with the ability to automate BPR using as much judgement as possible via computerised means, by introducing a degree of artificial intelligence. The technique should utilise cost estimation as the principal evaluation criterion.

5.1. Methodology

As described in Chapter 2, the generally accepted guidelines for BPR at the process level involve consideration of the following questions:

- Does the process contain steps which are repeated or redundant in certain circumstances?
- What would happen if these steps were not completed?
- Are there steps or series of steps which could be performed in parallel or combined with others?
- What are the risks of eliminating repeated or redundant steps or combining actions?
- Are there any problems with certain parts of the existing process which need to be addressed?

(Picot et al, 2004), (Grover and Malhotra, 1997), (Hammer and Champy, 1993)

The automated reengineering methodology attempts, where possible, to address these questions, normally undertaken by human experts, by performing the following steps:

- Search for and removal of duplicates (reengineering)
- Replacement of sections of process with one step (reengineering)
- Replacement of sections of process with sub process (reengineering)
- One for one replacements of process steps with steps of differing cost and duration (optimisation module)

The way in which this is achieved is now described.

Each process model generated is input to the reengineering algorithm, entitled ABPRS (Automated Business Process Reengineering System). A relational database contains all process data, including complete process models, sub-processes and process steps.

Many process steps and sub-processes may not be a part of the process models, but may be used as potential replacements for steps or sections of them during execution of ABPRS. The database is presented in Chapter 8. Both the methodology and the prototype software platform were developed in a modular and structured way. The modular stages involved in the reengineering are illustrated in the following program structure chart, Figure 26:

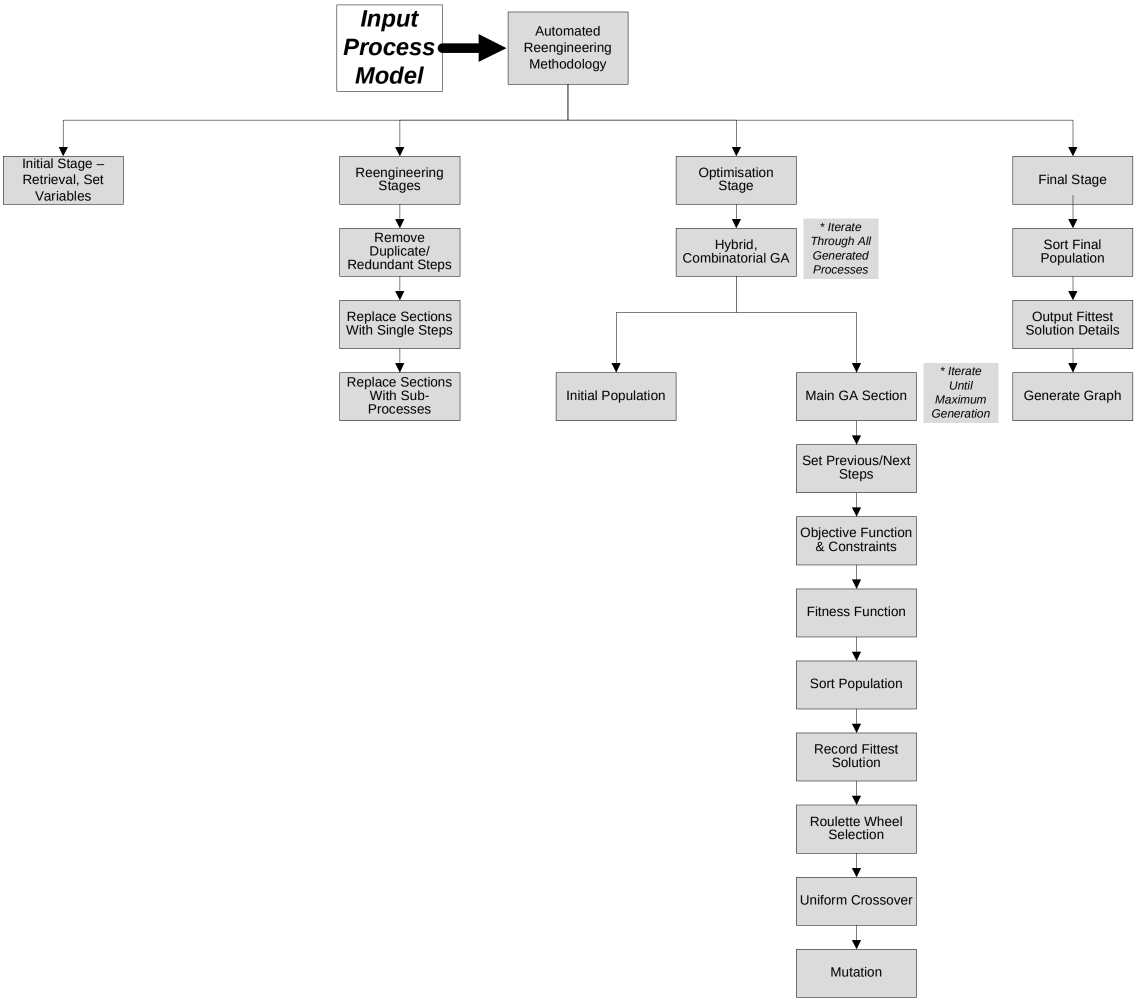

Input Process Model
Automated Reengineering Methodology
Initial Stage – Retrieval, Set Variables
Reengineering Stages
Remove Duplicate/ Redundant Steps
Replace Sections With Single Steps
Replace Sections With Sub-Processes
Optimisation Stage
Hybrid, Combinatorial GA
* Iterate Through All Generated Processes
Initial Population
Main GA Section
* Iterate Until Maximum Generation
Set Previous/Next Steps
Objective Function & Constraints
Fitness Function
Sort Population
Record Fittest Solution
Roulette Wheel Selection
Uniform Crossover
Mutation
Final Stage
Sort Final Population
Output Fittest Solution Details
Generate Graph

Figure 26: Automated Reengineering Methodology - Program Structure Chart

The following is a step by step description of how the automated re-engineering technique works. Some specific programming related details are excluded from the description, except where they are crucial to the understanding of its functionality. Some of the description is written in a form of pseudo-code. There are two underlying rules in force during the processing:

- Any process created in the 'Reengineering Stages' (see Figure 26) must not contain more than five steps in excess of the number contained in the existing process (this can be adjusted) – this allows for situations where the addition of more process steps may not necessarily indicate higher process cost or duration.

- Only feasible processes are created – infeasible processes cannot be evaluated or used.

5.1.1. Initial Stage

This part of the technique prepares an existing process for evaluation and potential reengineering. After selection of a process by the user (see Chapter 8), ABPRS is executed. All process data is retrieved from the database, variables are set, and a cost estimate of the existing process is performed.

```
Select a process to be optimised
Select a population size, number of generations, fraction of population to be kept and a mutation rate for the GA section of the technique
Calculate number of population members to survive each generation of GA section
        = (fraction kept * population size) rounded
Calculate number of mutations per generation of GA section
        = ((population size – 1) * mutation rate) rounded
Calculate number of matings per generation of GA section
        = ((population size – number to survive) / 2) rounded
Retrieve process from database
Retrieve all available process steps from database
Calculate overhead costs of the process
Record duration and resource costs for each process task
Record probabilities for each gateway
Estimate cost and duration of the process (ABC module)
```

5.1.2. Reengineering Stages

This part of the technique will produce at least 9 different processes, some or all of which may be the same, as follows:

1. Duplicates/redundant steps removed from existing process
2. Section of process replaced by a single step (forward search)
3. Section of process replaced by a single step (backward search)
4. Section of process 2 (above) replaced by sub-process (forward search)
5. Section of process 2 (above) replaced by sub-process (backward search)
6. Section of process 3 (above) replaced by sub-process (forward search)
7. Section of process 3 (above) replaced by sub-process (backward search)
8. Section of process output from 1 (above) replaced by sub-process (forward search)
9. Section of process output from 1 (above) replaced by sub-process (backward search)

A. Remove duplicate/redundant process steps

This part of the technique searches for possible duplicates or redundant steps within the existing process. It should be noted that the presence of duplicates may be intentional – for instance, sending two emails during the course of the process.

If one duplicate is removed that is superior to the other, in terms of cost and/or duration, the ‘Optimisation Stage’ (see Figure 26) of the technique should replace the inferior one with the superior one during its course.

This module uses input, output and constraint codes to determine the existence of duplicates, and whether or not they can be removed. If duplicates are found that can be removed, the next and previous step codes of connecting steps are reset to cater for the removal, and the removal takes place. The process is then re-evaluated.

```
Store existing process and output details
Begin at step after start node
Iterate until end node reached
        Record step’s input codes
        Examine each following steps' input codes
        If there is a match on any of the following steps' input codes, check the output codes_
                of this step
        If there is a match on any of the output codes, this is a duplicate –
                Check constraints of duplicate
                If no constraints -
                        Set Next/Previous step codes to cater for the removal of the duplicate
                        Remove the duplicate
Repeat

Objective Function: Estimate cost of the process (ABC module)
Constraint 1: Estimate duration of the process
Constraint 2: Estimate impact on Customer Satisfaction
Call Fitness Function
Store (new) process and output details
```

The following generic process, Figure 27, provides an example of this stage of the reengineering.

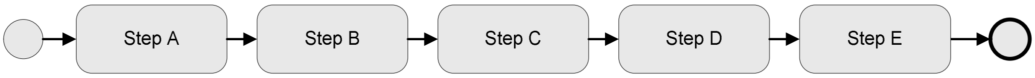


Figure 27: Remove Duplicates/Redundant Steps – Process Before

In the above serial process, Step C and Step E have one matching input code and one matching output code – so they will be identified as duplicate steps.

The sequence of events is as follows:

- Start
- Record Step A input codes
- Check against Step B – no match
- Check against Step C – no match
- Check against Step D – no match
- Check against Step E – no match
- Record Step B input codes
- Check against Step C – no match
- Check against Step D – no match
- Check against Step E – no match

- Record Step C input codes
- Check against Step D – no match
- Check against Step E – match
- Check Step E output codes against Step C output codes – match
- Check Step E constraint codes
- Set Next/Previous step codes
- Remove Step E
- Record Step D input codes
- Store process
- Output process
- End

This produces the following process, Figure 28:

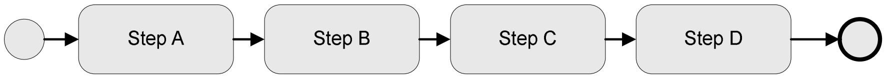


Figure 28: Remove Duplicates/Redundant Steps – Process After

B. Replace section of process with single process step

This part of the technique examines every process step held in the database. These are referred to as 'alternatives' and exclude the process steps that form the existing process. The aim is to replace more than one existing process step with a single step. This may potentially involve reconfiguration of the existing process. It is possible that the removal of process steps during a forward search may prevent another possible replacement from being tested, as illustrated in the example below. As such, this part of the technique is performed at least twice – once in a forwards direction and once in a reverse direction. Should this produce two different processes, as in the example below, the replacement steps used will be temporarily made unavailable and the procedure will be performed again on the input process until no change occurs (this caters for a situation where 3 or more potential replacement steps overlap – e.g. in the example below, if there was also a Step CDE, it would not be tested without this extra repetition). Input, output and constraint codes are again used to determine the suitability of alternatives, and whether or not existing process steps can be replaced. If replacements are made, next and previous step codes are reset to cater for the replacements and the process is re-evaluated.

```
Begin at step after start node
Iterate until two steps before end node reached, moving one step forwards in present_ process
    for each iteration
    Record step's input codes
    Examine every alternative process steps' input codes looking for a match
    If there is a match on any of the alternative steps' input codes –
        Record the matching alternative step's output codes
        Begin at following step in present process to one with input code match
        Iterate until end node
            Check present process steps' output codes
            If there is a match on output codes –
                Check constraints of step in present process with input code_
                    match
                Check constraints of step in present process with output code_
                    match
                If no constraints –
                    Set Next/Previous step and Flows In/Out codes to_
                        cater for the impending change
                    Replace the existing process steps with the single_
                        alternative step
        Repeat
Repeat

Objective Function: Estimate cost of the process (ABC module)
Constraint 1: Estimate duration of the process
Constraint 2: Estimate impact on Customer Satisfaction
Call Fitness Function
Store (new) process and output details
Retrieve previous process – becomes present process
Begin at step before end node
Iterate until two steps before start node reached, moving one step backwards in present_
    process for each iteration
    Record step's output codes
    Examine every alternative process steps' output codes looking for a match
    If there is a match on any of the alternative steps' output codes –
```

```
        Record the matching alternative step's input codes
        Begin at previous step in present process to one with output code match
        Iterate until start node
            Check present process steps' input codes
            If there is a match on input codes –
                Check constraints of step in present process with input code_
                    match
                Check constraints of step in present process with output code_
                    match
                If no constraints –
                    Set Next/Previous step and Flows In/Out codes to_
                        cater for the impending change
                    Replace the existing process steps with the single_
                        alternative step
        Repeat
Repeat

Objective Function: Estimate cost of the process (ABC module)
Constraint 1: Estimate duration of the process
Constraint 2: Estimate impact on Customer Satisfaction
Call Fitness Function
Store (new) process and output details
Compare 2 processes generated
If the same –
    end
If different – (this indicates that 2 different replacement steps have been used)
    Remove two replacement steps already used
    Retrieve input process
    Repeat all the above
```

The following generic process, Figure 29, provides an example of this stage of the reengineering.

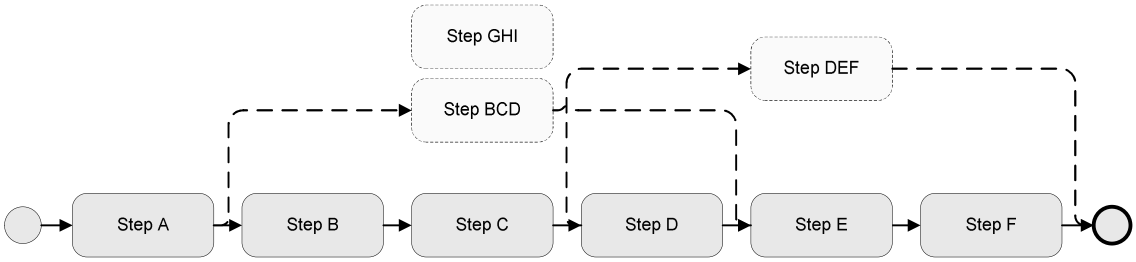


Figure 29: Section Replaced with Single Step – Process Before

In the serial process above, Steps B,C and D can be replaced by a single step from all the alternative steps available in the database – Step BCD (as illustrated with dashed lines) has one matching input code with Step B and one matching output code with Step D. Also, Steps D,E and F can be replaced by a single step from all the alternative steps available in the database – Step DEF (as illustrated with dashed lines) has one matching input code with Step D and one matching output code with Step F. It is assumed that there are only three other available steps in the database - Step DEF, Step GHI and Step BCD (in this order).

The sequence of events is as follows:

- Start
- Record Step A input codes
- Check against Step DEF – no match
- Check against Step GHI – no match
- Check against Step BCD – no match
- Record Step B input codes
- Check against Step DEF – no match
- Check against Step GHI – no match
- Check against Step BCD –match
- Record Step BCD output codes
- Check against Step C – no match
- Check against Step D – match
- Check Step B constraint codes
- Check Step D constraint codes
- Set Next/Previous/Flows step codes
- Replace Step B with Step BCD
- Remove Step C
- Remove Step D

- Check against Step E – no match
- Record Step E input codes
- Check against Step DEF – no match
- Check against Step GHI – no match
- Check against Step BCD – no match
- Store process
- Output process
- Retrieve previous process
- Record Step F output codes
- Check against Step DEF – match
- Record Step DEF input codes
- Check against Step E – no match
- Check against Step D – match
- Check Step F constraint codes
- Check Step D constraint codes
- Set Next/Previous/Flows step codes
- Replace Step F with Step DEF
- Remove Step E
- Remove Step D
- Check against Step C – no match
- Check against Step B – no match
- Check against Step A – no match
- Record Step C output codes
- Check against Step DEF – no match
- Check against Step GHI – no match
- Check against Step BCD – no match
- Record Step B output codes
- Check against Step DEF – no match
- Check against Step GHI – no match
- Check against Step BCD – no match
- Store process
- Output process
- If process 1 not = process 2
    - Remove first replacement step from database
    - Remove second replacement step from database
    - Go to start
- End

This produces the following two processes, Figures 30 and 31:

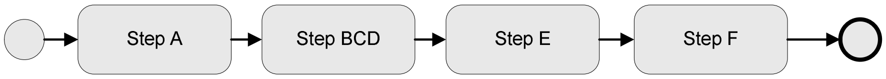


Figure 30: Section Replaced with Single Step – Process after Forward Search

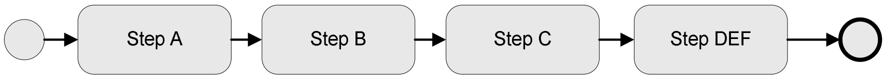


Figure 31: Section Replaced with Single Step – Process after Backward Search

C. Replace section of process with sub-process

This part of the technique examines every sub-process held in a library in the database. This can, as with the previous stage, allow reconfiguration of the existing process. This part of the technique is performed six times – once in a forwards direction and once in a reverse direction with the process created in the forward search of the previous part of the technique, once in a forwards direction and once in a reverse direction with the process created in the backward search of the previous part of the technique, and, to ensure the technique is as comprehensive as possible, once in a forwards direction and once in a reverse direction with the process created by the removal of duplicate/redundant steps. As in the previous part of the technique, the forwards/backwards search caters for the possibility of omitting certain replacements due to the removal of steps, and is repeated again if there are differences between a pair of processes created from the forwards/backwards searches. Again, input, output and constraint codes are used to determine suitability of sub-processes and whether or not existing parts of the process can be replaced. If replacements are made, and if the replacement sub-processes contain a maximum of 5 more steps than the section of the existing process to be replaced, next and previous step codes are reset to cater for the replacements, the replacements are made, and the process is re-evaluated.

```
Retrieve process created from previous forward search
Extract first sub-process from sub-process library
Iterate until all sub-processes have been checked
        Total number of steps in sub-process
        Record input codes to sub-process
        Begin at step after start node of present process
        Iterate until end node of present process
                Check if sub-process start node input codes match present existing process_
                        steps' input codes
                If input codes match –
                        Begin with same step in present process with matching input codes
                        Iterate until end node of present process
                                Check if sub-process end node output codes match present_
                                        process steps' output codes
                                Total number of steps from point where input codes match in_
                                        present process
                                If output codes match -
                                        Check if maximum of 5 more steps in sub-process
                                                than_ those to be potentially replaced in
                                                present_ process
                                        If maximum of 5 more steps in sub-process –
                                                Check constraints of first and last steps in_
                                                        present process to be replaced
                                                If no constraints -
                                                        Set Next/Previous step and Flows_
                                                                In/Out codes to cater for the_
                                                                impending change
                                                        Replace present process steps with_
                                                                sub-process (minus its start_
                                                                and end nodes)
                        Repeat
        Repeat
Repeat

Objective Function: Estimate cost of the process (ABC module)
Constraint 1: Estimate duration of the process
Constraint 2: Estimate impact on Customer Satisfaction
Call Fitness Function
Store (new) process and output details

Retrieve process created from previous forward search
Extract last sub-process from sub-process library
Iterate until all sub-processes have been checked
        Total number of steps in sub-process
```

```
    Record output codes to sub-process
    Begin at step before end node of present process
    Iterate until start node of existing process
        Check if sub-process end node output codes match present process steps'_
            output codes
        If output codes match –
            Begin with same step in present process with matching output codes
            Iterate until start node of present process
                Check if sub-process start node input codes match present_
                    process steps' input codes
                Total number of steps from point where output codes match_ in
                    present process
                If input codes match -
                    Check if less or same steps in sub-process than those_
                        to be potentially replaced in present process
                    If less or same steps in sub-process –
                        Check constraints of first and last steps in_
                            present process to be replaced
                        If no constraints -
                            Set Next/Previous step and Flows_
                                In/Out codes to cater for the_
                                impending change
                            Replace existing process steps with_
                                sub-process (minus its start_
                                and end nodes)
            Repeat
    Repeat
Repeat

Objective Function: Estimate cost of the process (ABC module)
Constraint 1: Estimate duration of the process
Constraint 2: Estimate impact on Customer Satisfaction
Call Fitness Function
Store (new) process and output details
Compare 2 processes generated
If the same –
    Next
If different – (this indicates that 2 different sub-processes have been used)
    Remove two sub-processes already used
    Retrieve input process
    Repeat all the above

Retrieve process created from previous backward search
Repeat all the above
Retrieve process created from removal of duplicate/redundant steps
Repeat all the above
```

In this example, the following sub-process, as shown in Figure 32, is available in the sub-process library.

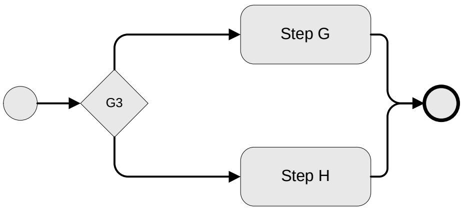


Figure 32: Replacement Sub-Process

The sub-process can be used as a replacement for the section of the present process Step A through to and including Step E, as illustrated below (Figure 33) with dashed lines.

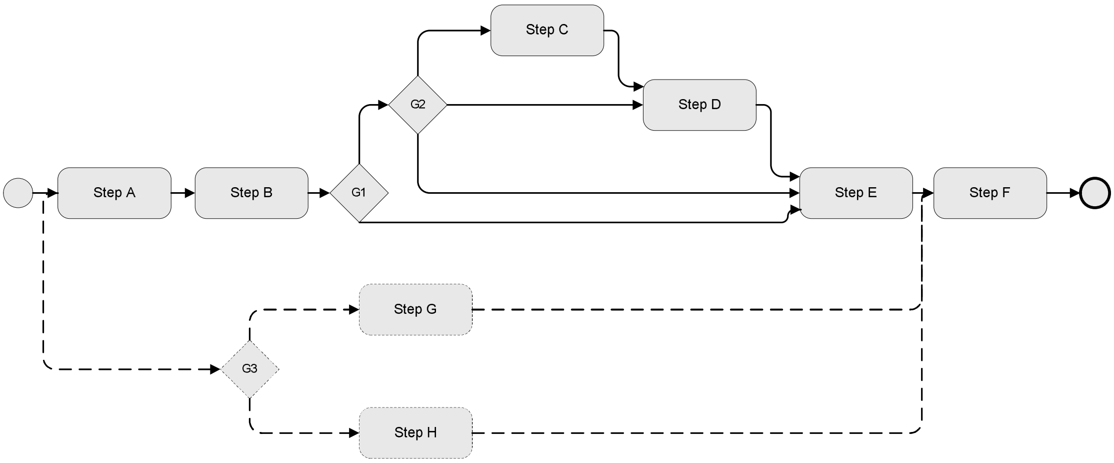


Figure 33: Section Replaced with Sub-Process – Before

The sequence of events is as follows:

- Start
- Retrieve process created from previous forward search
- Record input codes to sub-process
- Check against Step A – match
- Record number of steps in sub-process
- Record output codes to sub-process
- Check against Step A – no match
- Add to step total
- Check against Step B – no match
- Add to step total
- Check against Step G1 – no match
- Add to step total
- Check against Step G2 – no match
- Add to step total
- Check against Step C – no match
- Add to step total
- Check against Step D – no match
- Add to step total
- Check against Step E – match
- Add to step total
- Check sub-process shorter than section to be replaced
- Check constraints of first/last steps to be replaced
- Set Next/Previous/Flows step codes
- Replace Step A with Step G3
- Replace Step B with Step G
- Replace Step G1 with Step H
- Remove Step G2
- Remove Step C
- Remove Step D
- Remove Step E
- Check against Step F – no match
- Store process
- Output process
- Record output codes to sub-process
- Check against Step F – no match
- Check against Step E – match
- Record number of steps in sub-process
- Record input codes to sub-process
- Check against Step D – no match
- Add to step total
- Check against Step C – no match

- Add to step total
- Check against Step G2 – no match
- Add to step total
- Check against Step G1 – no match
- Add to step total
- Check against Step C – no match
- Add to step total
- Check against Step B – no match
- Add to step total
- Check against Step A – match
- Add to step total
- Check sub-process shorter than section to be replaced
- Check constraints of first/last steps to be replaced
- Set Next/Previous/Flows step codes
- Replace Step E with Step H
- Replace Step D with Step G
- Replace Step C with Step G3
- Remove Step G2
- Remove Step G1
- Remove Step B
- Remove Step A
- Store process
- Output process
- If process 1 not = process 2
    - Remove first sub-process from database
    - Remove second sub-process from database
    - Go to start
- End
- Repeat with process from previous backward search
- Repeat with process from duplicates/redundant steps

This produces the following process (Figure 34):

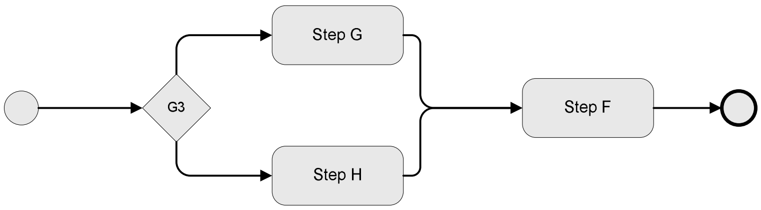


Figure 34: Section Replaced with Sub-Process – After

5.1.3. Optimisation Stage

There are now at least 9 processes available as follows:

1. Duplicates/redundant steps removed from existing process
2. Section of process replaced by a single step (forward search)
3. Section of process replaced by a single step (backward search)
4. Section of process 2 (above) replaced by sub-process (forward search)
5. Section of process 2 (above) replaced by sub-process (backward search)
6. Section of process 3 (above) replaced by sub-process (forward search)
7. Section of process 3 (above) replaced by sub-process (backward search)
8. Section of process output from 1 (above) replaced by sub-process (forward search)
9. Section of process output from 1 (above) replaced by sub-process (backward search)

A check is made on the processes for differences – it is possible that some or all of them may be exactly the same. No duplicate processes are used in the next stages of the technique.

The following part of the technique, which uses the GA operators to maximise the search space, is performed for each unique process (1 to *n* times – according to number of generations selected). The end result is between 1 and *n* fittest solutions, of which the fittest one is selected. This is a hybrid, combinatorial (i.e. characterised by a finite number of feasible solutions, using heuristics (Gen and Cheng, 1997)) algorithm, operating within the constraint of only creating feasible processes. In order to do this, only available process data is used, and it is ordered in such a way that only creates feasible processes. As such, some of the random, or stochastic, nature of GA is sacrificed. This is described in the sections Initial Population, Uniform Crossover and Mutation. Nevertheless, the GA still provides a means of maximising the available search space, and generating and analysing potentially very large numbers of processes. Figure 35 illustrates the flow of the optimisation module in which each unique process generated by the reengineering stages serves as an input:

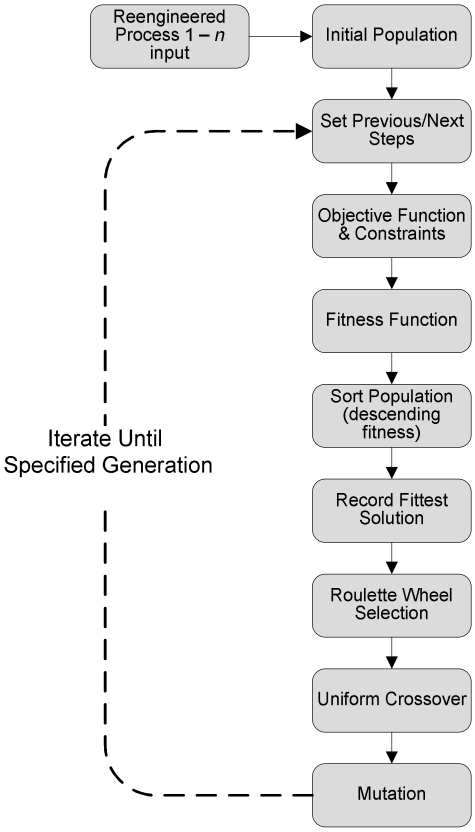

Reengineered Process 1 – *n* input
Initial Population
Set Previous/Next Steps
Objective Function & Constraints
Fitness Function
Sort Population (descending fitness)
Record Fittest Solution
Roulette Wheel Selection
Uniform Crossover
Mutation
Iterate Until Specified Generation

Figure 35: Optimisation Stage Flowchart

Figure 35 depicts a simple GA. GA works with populations of ‘chromosomes’ through which each potential solution is represented – this is often a binary representation but can also take other forms such as the representation of processes in this research. Each chromosome consists of a number of ‘genes’ which may be zeroes and ones, but in this research are the 174 digit process step representations (e.g. tasks and gateways) described in Chapter 4 that, when strung together, represent the entire process model. The position of each gene in the chromosome is referred to as its ‘locus’.

A. Initial population

This part of the technique examines every available process step in the database. Potential replacements for steps in the input process are identified via input and output codes, and the initial population is created from the input process plus identified alternative steps. All process chromosomes created must represent feasible processes.

```
Iterate until every step in input process checked
    Iterate until every alternative checked
        If input codes match alternative input codes –
            If output codes match alternative output codes –
                Store alternative and its position in process
    Repeat
Repeat
```

If no alternatives for any process steps have been identified at this point, it is unnecessary to continue any further with this particular input process – so the next process is moved to.

An array is created from the process data as follows:

- First row - input process.

- Second row - all those steps that have alternatives are replaced with their first available alternative. Those without alternatives are left blank.

- Third row - all those steps that have alternatives are replaced with their second available alternative. Those without alternatives are left blank.

- This continues until all possible alternatives have been used – 50 alternatives per process step are catered for (e.g. numerous external suppliers, courier services, technologies, etc.).

  For example, using 2 character process step codes instead of 174 digit codes:

Row 1 =	a1	b1	c1	d1	e1	f1
Row 2 =	-	b2	-	d2	-	f2
Row 3 =	-	b3	-	-	-	f3
Row 4 =	-	-	-	-	-	f4

The initial population is then created randomly from this array – for example, column 2 of row 12 becomes gene 2 of chromosome *n*, column 7 of row 4 becomes gene 7 of chromosome *n*.

```
Record number of rows in array
Iterate until population size reached
    Iterate until number of steps in process
        Generate random number (1 – number of rows in array)
        If not blank –
            Insert gene from array(random number, step number) in_
                chromosome(population number, step number)
            Set next/previous step codes
        If blank –
            Generate another random number
    Add 1 to step number
    Repeat
Add 1 to population number
Repeat
```

The idea behind this is that the GA could create every possible combination using all alternatives available during its course. Even if all alternatives do not appear in the initial population, they may be selected via Mutation.

The following sections, ‘B Previous and next step codes’ through to ‘H Mutation’, are repeated until the specified number of generations has been reached.

B. Previous and next step codes

Previous and Next Step codes are essential to the evaluation of the process chromosome and its translation into a process diagram - therefore they must be set correctly after every change to the chromosome. The method of setting them correctly involves checking all identified alternatives’ IDs along with the original process step ID and ascertaining which ID to use. The following provides a simple example of this situation (again with example 2 character step codes):

- A process contains the steps A1, B1 and C1 in sequence.

  Alternative steps A2 and C2 are identified as potential replacements for steps A1 and C1 respectively.

  The existing process's previous and next step codes are:

Step A1:	Previous Step Code(1) = Start Node
	Next Step Code(10) = B1
Step B1:	Previous Step Code(1) = A1
	Next Step Code(1) = C1
Step C1:	Previous Step Code(1) = B1
	Next Step Code(1) = End node

Then, during execution of the optimisation module, steps A1 and C1 are replaced with steps A2 and C2.

When step A1 is replaced by A2, any previous or later step that has a next or previous step of A1 is analysed – in this case, step B1. This is based on a search of all alternatives for that step (A1 and A2 in this case) and, once the alternative used as a replacement is verified (A2), the appropriate changes to the next/previous step codes in previous or later steps takes place (B1). Exactly the same applies for step C1. Thus, the reengineered process's codes are reset as follows:

Step A2:	Previous Step Code(1) = Start Node
	Next Step Code(10) = B1
Step B1:	Previous Step Code(1) = A2
	Next Step Code(1) = C2
Step C2:	Previous Step Code(1) = B1
	Next Step Code(1) = End node

C. Objective function, constraints and fitness function

The objective function for estimating process cost (see Chapter 6) is executed and the constraints which involve the measurement of process duration (see Chapter 6) and detection of impact on customer satisfaction (see Chapter 7) are applied. This is followed by the fitness function for allocating a fitness to each chromosome, achieved using Delta Analysis in which the existing process is used as a comparison (van der Aalst et al, 2004) - the objectives, constraints and fitness function are described in more detail in Chapter 8. The above is performed on each member of the population. This involves subtracting the estimated cost of the process being evaluated from the estimated cost of the existing process and using the generated value as the fitness 'score' – a positive value indicates a lower cost, and vice versa. It is possible that this score will be zeroised in cases of excessive process duration or negative impact on customer satisfaction, or it may be increased in cases of positive impact (see Chapter 7 and 8 for details).

D. Sort population

The population is sorted by descending fitness (highest fitness = population member 1)

E. Record fittest solution

The chromosome with the highest fitness is stored in a table

F. Roulette wheel selection

Standard Roulette Wheel Selection is used to allow fitter solutions a greater chance of selection in the following crossover and mutation (Goldberg, 1989). A 'from' and 'to' value is attached to each chromosome – later, a random number between 0 and 100 is generated and, if the number lies between the 'from' and 'to' values of a particular chromosome, it will be selected – the higher the fitness, the larger the gap between the 'from' and 'to' value.

```
Check for zero or negative value in lowest fitness chromosome
(for allocation of sections of roulette wheel, lowest fitness must be at least 1)
 If  zero – add 1 to every fitness score in population
If negative – score = (score * -1) + 1 and add to every fitness score in population
Calculate total fitness score of population
Set first chromosome ‘from’ value to 0
Calculate ‘to’ value = (fitness score/total fitness score) * 100
Iterate until all population allocated ‘from’ and ‘to’ values
        ‘from’ value = previous chromosome ‘to’ value
        ‘to’ value = ((fitness score/total fitness score) * 100) + ‘from’ value
Repeat

Iterate until calculated number of matings
        Generate random number 0-100
        Select chromosome with ‘from’ and ‘to’ values containing random number as ‘mother’_
                chromosome
        Generate random number 0-100
        Select chromosome with ‘from’ and ‘to’ values containing random number as ‘father’_
                chromosome
Repeat

Iterate until calculated number of matings
        Set indexes of children chromosomes = number to survive + 2
        (If 10 are to survive out of a population of 20, indexes will be 12,14,16,18,20)
Repeat
```

G. Uniform crossover

Uniform crossover is used as it causes a high level of disruption to each chromosome and is arguably superior for certain problems (Man et al, 1997) – initial experimentation with single-point crossover verified this. It also allows the locus of each gene to remain unchanged, thus maintaining the feasibility of each process chromosome. A ‘binary mask’ is created containing a string of zeroes and ones, randomly generated with a 50/50 chance of either value. Each value relates to a gene in the chromosome: one indicates mother chromosome, zero indicates father chromosome.

```
Iterate until binary mask array index = number of genes per chromosome
    Randomly generate 0 or 1
    Store in array
    Add 1 to index
Repeat

Iterate until calculated number of matings
    Iterate until end of process chromosome
    (start with first gene in process chromosome, adding one for each iteration)
        If binary mask = 1 –
            mother gene(mother chromosome index) replaces children_
                gene(children chromosome index)
            father gene(father chromosome index) replaces children gene(children_
                chromosome index + 1)
        If binary mask = 0 –
            father gene(father chromosome index) replaces children gene(children_
                chromosome index)
            mother gene(mother chromosome index) replaces children_
                gene(children chromosome index + 1)
    Repeat
Repeat
```

e.g.							
	Mother Chromosome	A1	B1	C1	D1	E1	F1
	Father Chromosome	G1	H1	I1	J1	K1	L1
	Mask	0	1	1	0	0	1
	Child Chromosome 1	G1	B1	C1	J1	K1	F1
	Child Chromosome 2	A1	H1	I1	D1	E1	L1

The number of crossovers that take place per generation, or crossover probability, is determined by the user input parameters (see Chapter 8 and 9 for details). This is calculated as follows:

Number of Crossovers =

*(Population Size – ((Fraction of Population Kept * Population Size) Rounded)) / 2 Rounded*

Uniform Crossover takes place between process chromosomes which are selected using

Roulette Wheel Selection – the higher the allocated fitness of a process chromosome, the higher its chance of selection for crossover. These are the 'parents' – *n* (number of crossovers) 'mothers' and *n* 'fathers' are selected. The resultant 'offspring' or 'children' are placed in the section of the population based on the 'fraction of population kept' value supplied – they are placed in the section of the population not to be kept (e.g. if fraction of population kept = 0.5 and population size = 100, the offspring are placed in population members 51 to 100).

H. Mutation

As with uniform crossover, mutation must not change the locus of any genes. Genes are only selected for mutation where alternatives have been identified for that locus in the chromosome – or position in the process. If no alternatives are available, mutation would be pointless as the gene would be replaced with its exact same value.

```
Iterate until number of mutations reached
    Generate random number 2-population size (fittest chromosome is not subject to mutation)
    Index of chromosome to mutate = random number
    Generate random number 1- number of genes with alternatives
    Index of gene to mutate = random number
    Randomly select gene from identified alternatives
    Replace gene(chromosome index, gene index) with randomly selected gene
Repeat
```

The number of mutations that take place per generation, or mutation probability, is determined by the user input parameters (see Chapter 8 and 9 for details). This is calculated as follows:

Number of Mutations =
*((Population Size -1) * Mutation Rate) Rounded*

Mutation takes place *n* (number of mutations) times per generation. Chromosomes are randomly selected from the entire population, and a gene from each of these chromosomes is also randomly selected. For each mutation, a gene (or process step) is then selected from potential replacement steps for existing process steps – these may or may not be present in the current population. Only steps that have been identified as potential replacements for the selected gene can be used in order to maintain the feasibility of the process chromosome (i.e. the locus cannot change).

5.1.4. Final Stage

All unique processes generated by the reengineering stages of the technique have now been input one at a time to the optimisation stage (see Figures 26 and 35). Therefore, a number of 'optimum solutions' have now been generated and stored – one for each of the unique processes input (which may or may not have changed as a result of the optimisation stage). The solutions are sorted in order of fitness, based on the fitness score they have been previously allocated, and details of the optimum are output, plus details from the GA showing rises and falls in process cost during its execution:

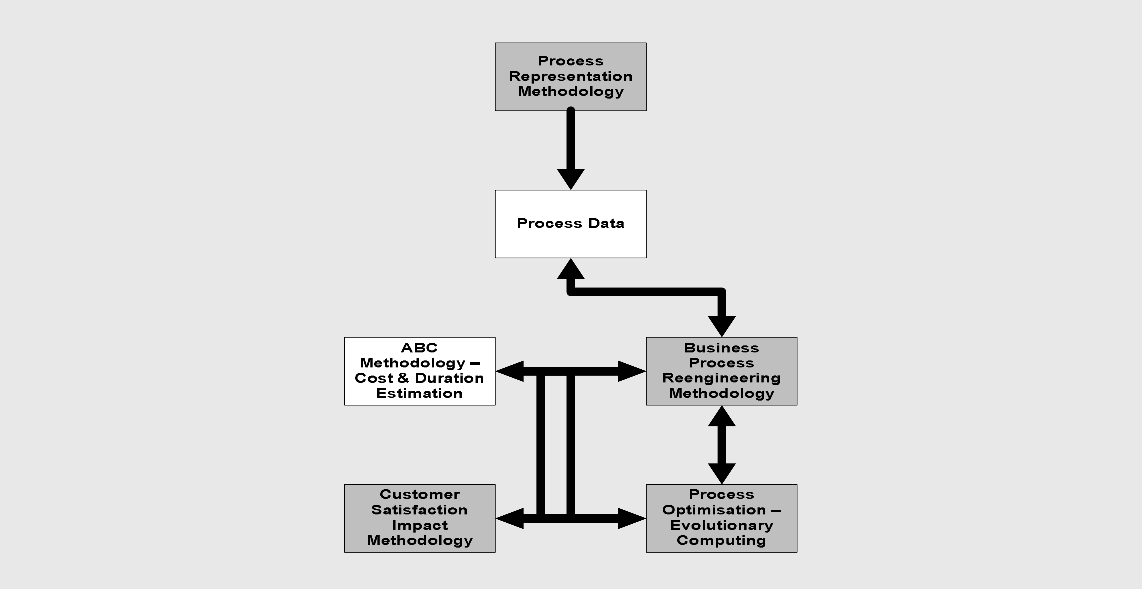
- Sort processes in order of fitness (i.e. fitness score descending).
- Select process with highest score as the 'optimum solution'.
- Output full details of fittest process via report (see Chapter 8)
- Generate chart detailing process optimisation (see Chapter 8)

5.2. Summary

This chapter has described the detailed functionality of the automated reengineering methodology: how it uses all available process data in order to try and apply the guidelines of BPR, and how it incorporates GA operators in an attempt to maximise the available search space and potentially generate every feasible process available - which could be used as an optimal replacement for the existing process. Earlier experimental work attempted to configure processes out of random selections of process steps. This failed, mainly due to the free-standing nature of many of the steps in each process – they are unconstrained and could appear in any part of the process. Only constrained parts of the process could be configured – for instance, gateways, which must be preceded by a specific step and must be followed by one or more specific steps, which are detailed in the code for the gateway.

The objective, process cost reduction, is covered in the next chapter in which the cost estimating methodology is described. This methodology is the principal part of the ‘Objective Function & Constraints’ module of the overall reengineering methodology (See Figure 26).

6. Probability Driven Activity Based Cost Estimation of Business Processes


Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

The methodology presented in this chapter attempts to address the research gap identified in Section 2.10: a requirement to introduce another level of detail to an ABC estimate when applied to an automated reengineering technique for service sector processes - by developing a probability (of a process following certain paths) driven ‘Activity and Task Based Costing’ methodology.

Due to its link with activities, ABC immediately appeared to be a suitable option for use in this research project: there is a direct association between activities and business processes, as a business process is essentially an activity or a series of activities. The requirement to break down business processes and reengineer them based on cost criteria is entirely compatible with this technique – in this research project, a structure is proposed in which each task within a process has direct resource costs associated with it and each activity has overhead costs which are rates based on process duration. This direct resource to task relationship structure is proposed to allow for differentiation between:

- Reengineered processes that use the same resources as the original process, but in different degrees due to:

  - Changes in the probability of the resources being used
  - Replacement of tasks with ones of different durations

- Reengineered processes that use some different resources than the original process, and which therefore may have different costs, but that produce the same output.

6.1. Industry Survey

In order to assist in developing the ABC methodology, an industry survey was conducted with companies involved in banking and telecommunications: HSBC Bank, Clydesdale Bank and three British Telecom business units (UK and USA based). One respondent was selected from each organisation (five in total) who was best qualified to complete the survey. The job titles of the respondents varied between Cost Analyst, Financial Controller and Principal Commercial Manager – with experience of between 2 and 13 years. The self-completion survey technique was used

The main purpose of the survey was to identify variations in the way in which these companies practised business process cost estimation and, if they used an ABC methodology, how they had implemented it (i.e. the structure) and how they used it (see Appendix A for questionnaire). The questions were designed to extract as much information as possible from each organisation about their cost estimating practices, whether ABC or not, and covered the following areas:

- The process of predicting the cost of business processes, detailing information used, the level of detail, outputs and data collection methods, and the main challenges with this process
- All software used to assist in cost estimation
- Types of costs and drivers
- Any positive and negative aspects of ABC where used for cost estimating
- Evaluation of the accuracy of estimates

All answers were compared - it was found that each company has a different way of dealing with its cost estimating process depending on the nature of its business, but also some of the companies shared certain similarities, notably in their examination of cost data at a relatively high level (i.e. departmental). Only one of the five companies adhered quite strictly to ABC theory - this flexibility proved to be valuable information. Some of the answers provided were useful in determining exactly how to construct the cost estimating facility – for instance, the cost drivers established are per transaction for direct resources and per unit of time for overheads. The following excerpts from the survey provide an indication of how the companies involved achieve their cost estimates:

- "…Capture all elements within the business process to include: employee salary, time the process takes to complete and any cost related to tools needed to complete the process. The analysis would determine the cost of the process by instance and this could be put into terms of measurement such as cost per action, per hour, per employee, per month cost and year on year cost for the business process"

- "…we identify all the activities carried out within the department that performs this business process, whether driven by systems or staff, and measure dependencies and outputs using stats and other volumetrics available. The total cost incurred by said department is then assigned to each activity, in the right proportion, by using the aforementioned measures"

- "…costs per department and account (expense type) are obtained from the General Ledger system. Then, these are mapped to the resources that perform all the activities in each department, using a method or driver. Activities can be clustered according to the process"

- "…Prediction is carried out mainly through trend / scenario analysis. We use various cost allocation tools to obtain the required data for this analysis. These are Metify and Oros. We have started to use PeopleSoft ABM as well"

- "…We carry out staff interviews, issue surveys and data collection forms, and access relevant data available on source systems. All of these provide information on business areas, accounts, activities, products, customers, channels and cost drivers"

- "…ABC allows us to trace the costs, incurred when providing products and services to our customers through the various channels, back to the activities carried out, the resources that performed those activities and the business areas those resources belong to. Therefore, it is a very thorough way of understanding costs in an organisation. All of these variables (business area, expense type, activity, product, customer, channel) are linked in the ABC methodology"

The above answers illustrate company specific means of gathering cost data and subsequently allocating costs to activities and/or processes. Despite the differences in the methods used, all companies endeavour to achieve the same result – an estimate, as accurate as possible, of the cost of running their various business processes. The level at which these companies capture their cost data is generally departmental or account. The aggregation of costs at this relatively high level, or even at the process level described in the first excerpt above, would make recognition of variations in probability of resource usage difficult to achieve – however, when making an instant evaluation of a selection of reengineered processes which use the same resources, but in differing degrees, it is necessary to examine costs at a lower level, such as the methodology presented in this chapter suggests. It is certain that changes in amount of resource usage for a department would eventually be detected, and allocated costs per activity within the department adjusted accordingly, but an immediate detection is required to effectively differentiate between and evaluate identical resource usage processes when applying the automated reengineering presented in this thesis.

Information from the survey relevant to this project and which assisted in the development of the technique, either directly or indirectly, is listed below in Table 12.

Table 12: Industry Survey - Relevant Points

RELEVANT POINTS	SUMMARY
Costs included in Estimate (fixed, variable, direct, indirect, support, customer facing)	Initial Preparation Administration Ongoing updates Maintenance of work instructions Travel Employee salaries Equipment
Level of cost data collection	Departmental Account Process
Estimation enablers including software packages	Business cases/historical data/ benchmarks Trend/scenario analysis Economic statistics/surveys Procurement supported bids ARIS PPMS Metify Oros Peoplesoft ABM Oracle/Tomcat/Excel
Measurement units	Cost per action/activity/run Cost per hour/month/year Cost per employee

6.2. ABC Forum

Once an approximate framework for a cost estimating methodology had been designed, a question was posed on an ABC forum (12MANAGE, 2008) for verification purposes. The question was as follows:

Title: Conditional Resources in ABC

Question: “How does ABC cater for estimating the cost of business processes which may take many different routes depending on the outcome of decisions or events – for instance, a resource may only be used once in every hundred runs of the process, therefore is only 1% of the resource cost included in the estimate?”

There were a total of thirteen responses to this question: twelve did not accept that only 1% of the resource cost should be included, indicating cost aggregation at a higher level (e.g. departmental); one agreed that, when assigning costs to an activity, online transaction processing (OLTP) data would be used to establish the amount of resource usage, and adjusted to allow for this 1% utilisation.

However, when evaluating reengineered processes whose amount of resource usage may have changed due to variations in probability of usage, it would be impossible to differentiate between the original process and the reengineered one without a means of re-estimating the process cost based on amount of resource usage. Also, the duration of the process may change but amount of resource usage remain the same, although this change in process duration would be detected by a standard ABC methodology. Plus, amount of resource usage and duration may remain the same, but the resources used may change, still producing the same output but at a different cost. Therefore, the responses from the forum, along with the survey, established a requirement for adding a further level of detail to the estimation of the cost of a business process – particularly a variable business process.

6.3. ABC Methodology Developed

The principal steps that led to the development of the ABC methodology were as follows:

- Literature review and case study led to the understanding that business processes, due to human interaction, can follow different paths when they are executed based on the outcome of decisions or events.

- The survey highlighted an organisation-dependent degree of flexibility in ABC, indicated the level at which cost data is collected and examined, and also provided cost drivers for use by the proposed methodology.

- The forum verified the requirement for an extra level of detail in the ABC methodology - to allow for changes in probability of resource usage when evaluating reengineered processes. This should address the research gap identified in Section 2.10 - a requirement to introduce another level of detail to an ABC estimate when applied to an automated reengineering technique for service sector processes, by developing a probability (of a process following certain paths) driven 'Activity and Task Based Costing' methodology.

In a typical ABC structure, each activity comprises a number of tasks. Tasks are one type of process step (or sub-transaction). Table 13 shows an example activity with the resources allocated to each task.

Table 13: Example Tasks and Resources

ACTIVITY – PLACE ORDER	RESOURCES
Input Customer Name	Internal Customer RDB
Input Customer Address	Internal Customer RDB
Input Payment Details	Internal Customer RDB
Check Postal Code	External Postcode Check DB Server
Input Product Details	Internal Stock RDB
Order Product	Internal Orders RDB
Generate Order Number	Internal Orders RDB
Request Product Delivery	External Supplier DB
Authorise Payment	External Payment Authorisation Secure Server
Email Confirmation	Email Account

With the ABC methodology proposed in this project, on top of these specific activity-task-resource relationships are 4 additional/overhead costs, supplied by the industrial sponsor, that are associated with each of the processes. They are apportioned to each activity centre as a rate per unit of time and then divided equally among all the activities in the activity centre. They are broken down, as illustrated below in Figure 36.

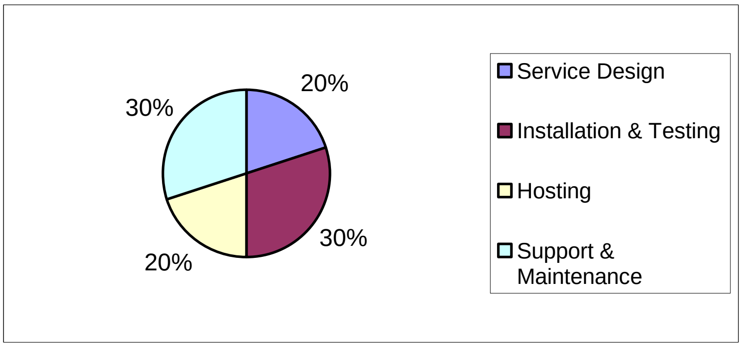


Figure 36: Additional/Overhead Cost Allocation (as supplied by industrial sponsor)

These rates are multiplied by estimated process duration for each activity/business process and added to the direct resource cost estimate. Process duration is estimated in exactly the same way as cost – each task has a duration attached to it.

All cost (and duration) data is held in a relational database, described in detail in Chapter 8, where it can be retrieved and used for process evaluation. A simplified snapshot of the relevant cost-related database tables plus example data is shown in Figure 37.

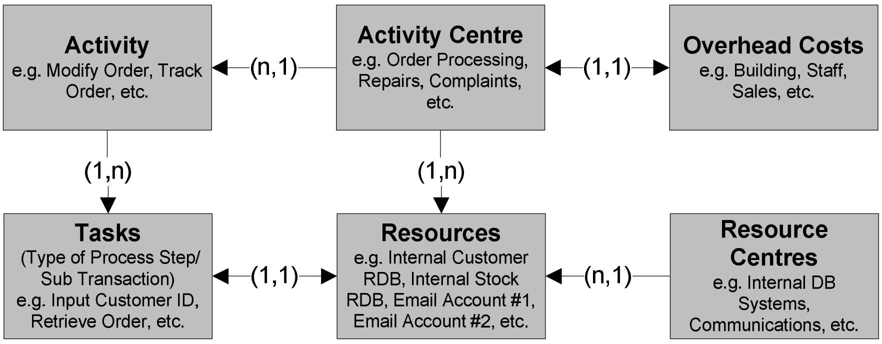

Activity
e.g. Modify Order, Track Order, etc.
(n,1)
Activity Centre
e.g. Order Processing, Repairs, Complaints, etc.
(1,1)
Overhead Costs
e.g. Building, Staff, Sales, etc.
(1,n)
(1,n)
Tasks
(Type of Process Step/ Sub Transaction)
e.g. Input Customer ID, Retrieve Order, etc.
(1,1)
Resources
e.g. Internal Customer RDB, Internal Stock RDB, Email Account #1, Email Account #2, etc.
(n,1)
Resource Centres
e.g. Internal DB Systems, Communications, etc.

Figure 37: Cost Data

As mentioned previously, many business process models, when executed, can follow one of a number of routes. It follows, therefore, that the process can have a number of different costs. The research presented here is an attempt to generate an accurate ABC estimate of any business process that can follow more than one route. It can also be applied to purely serial processes.

During execution of the automated reengineering technique, the ABC methodology is used to evaluate potentially very large numbers of different processes, all of which have the same input and produce the same output, but may vary significantly in their constituent process steps. As such, it was important to ensure that the functionality of the ABC methodology was as efficient as possible in order to maintain as short a run-time as possible.

The following example in Figure 38 shows a business process model, based on a real-life ARIS process model, provided by the sponsors, of a credit check process.

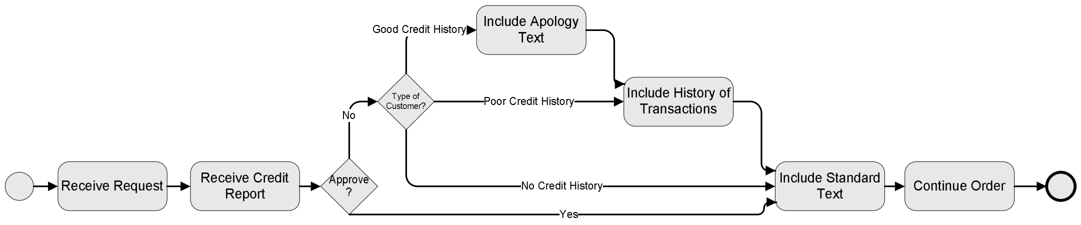


Figure 38: Credit Check Process

This process can take one of four different routes based on decisions generated. These decisions are the outward flows from ‘Gateways’, the diamond shaped notations, which appear at the sub-transaction, or process step level of the taxonomy, as described in Chapter 4. Gateways can branch to one of five different routes based on events or decisions (as in this case) - the methodology caters for a maximum of five outward flows from a gateway. They can also fork to between two and five parallel routes. It is the branching gateways that have an effect on the cost (and duration) of a process. Each outward flow from a branching gateway has a probability attached to it – ideally, these probabilities will be based on past process execution data and should be updated periodically.

Tasks, which also occur at the sub-transaction/process step level of the taxonomy, are the process steps which have resource costs attached to them. Other types of steps appear in a process model but it is tasks, along with gateways, which determine the cost estimate of a process.

The following, Figure 39, shows the credit check process with resource costs/duration and probabilities. All resource costs displayed are simply cost units and are not intended to accurately portray the real cost of this process. The numbers, 1, 2, 3 and 4, illustrate the four paths that this process can follow.

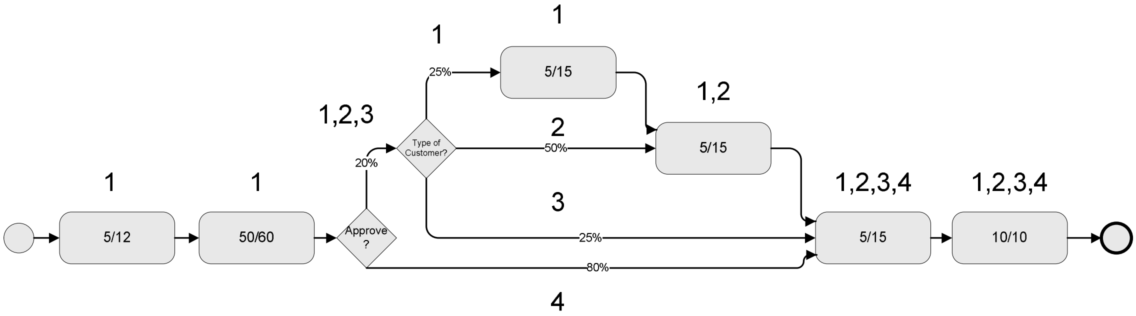


Figure 39: Credit Check Process – Costs & Probabilities

The technique that has been developed to estimate the process costs and durations navigates through all possible routes while stacking up probabilities as gateways are encountered. When the end node is reached on a particular route, it backtracks to the previous gateway and follows the next flow – this is repeated until all possible flows have been exhausted (i.e. all flows from the first gateway encountered have been followed through). The following table, Table 14, represents the virtual array that is created as a result of this. This array provides an indication of the steps involved in the automated methodology and the order in which they happen. Each row represents a possible route that the process can take – plus one per gateway with sequence flow loop (see below), and minus one per parallel flow (except for the first of the set of parallel flows). There is no set formula for calculating the number of process routes based on the number of gateways and their outward flows – this is dependant on the configuration of the process model.

Table 14: Credit Check Process – Cost Estimate

PATH	RESOURCE COSTS/DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/DURATION	CUMULATIVE PROBABILITY	TOTAL
1	55/72	20%	-	25%	25/55	5.00%	56.250/74.750
2	-	20%	-	50%	20/40	10.00%	2.000/4.000
3	-	20%	-	25%	15/25	5.00%	0.750/1.250
4	-	80%	15/25	-	-	80.00%	12.000/20.000
						100.00%	**71.000/100.000**

The totals for each path are as follows:

- Path 1 Resource Cost 55 + (0.2 * (0.25 * 25)) = 56.250

Path 1 Duration	72 + (0.2 * (0.25 * 55))	= 74.750
Path 2 Resource Cost	0.2 * (0.5 * 20)	= 2.000
Path 2 Duration	0.2 * (0.5 * 40)	= 4.000
Path 3 Resource Cost	0.2 * (0.25 * 15)	= 0.750
Path 3 Duration	0.2 * (0.25 * 25)	= 1.250
Path 4 Resource Cost	0.8 * 15	= 12.000
Path 4 Duration	0.8 * 25	= 20.000

This translates into the calculation:

- Resource Cost = 55 + (0.2 * ((0.25 * 25) + (0.5 * 20) + (0.25 * 15))) + (0.8 * 15)
- Duration = 72 + (0.2 * ((0.25 * 55) + (0.5 * 40) + (0.25 * 25))) + (0.8 * 25)

Therefore, the resource cost estimate for this process is 71 and the duration estimate is 100.

The duration estimate is then multiplied by the additional/overhead costs allocated to this process (i.e. the activities within the process) and this figure is aggregated with the resource cost estimate.

The next sections, 6.3.1 and 6.3.2., describe how the technique deals with the more complicated situations created by process loops and forks.

6.3.1. Sequence Flow Loops

The above example of the credit check process is relatively straight forward. However, this becomes more complicated when gateways that contain a sequence flow loop are encountered. A sequence flow loop is a situation where one of a gateway's outward flows involves a repetition of a previous part of the process as a result of an unsuccessful outcome from that part of the process – there are only two flows from this type of gateway: one forward flow and one backward flow. These types of loop are typical of variable, service based business processes in which decisions are made by the customer. The following, Figure 40, shows a simple sequence flow loop:

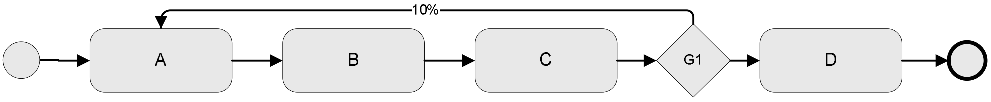


Figure 40: Sequence Flow Loop – Type 1

There is a 10% probability that gateway G1 will flow back to A and repeat A, B and C. There is a 90% probability that it will not. It would be impossible to allow for an unspecified number of iterations of this sequence flow loop in the cost estimate and so it was decided to cater for one iteration only – that there is a 10% probability that steps A, B, C will be iterated once as a result of the sequence flow loop and then, when gateway G1 is reached again, the flow will continue to D. Based on past process execution data, the sequence flow probability can be fine tuned to allow for sequence flow loops that, on average, are iterated more or less than a 10% probability of one iteration.

The above example is relatively simple because the flow backwards covers a purely serial section of the process. As such, when the 10% probability has been satisfied in the calculation, the gateway is effectively 'cancelled out' and the other probability can be set to 100%. Previous and later probabilities are now unaffected by this gateway and it is not added to the stack of probabilities. The cost estimate of this process is therefore as follows:

A + B + C + (0.1 * (A + B + C)) + D

However, this situation becomes more complicated when the sequence flow loop involves the repetition of a part of the process that contains previous gateways – it becomes linked to these other gateways' probabilities and becomes a part of other decisions. In this case, the alternative probability in the sequence flow loop gateway remains intact after the loop has been catered for – resetting it to 100% would render the final cost estimate inaccurate. Also, only the steps in the process that lead directly to the sequence flow loop gateway must be iterated – following other routes would inevitably lead to an even more inaccurate estimate and/or lead to the technique effectively 'getting lost' in its attempt to iterate the sequence flow loop once. This situation can be illustrated in the following generic example (Figure 41), showing resource costs and probabilities only:

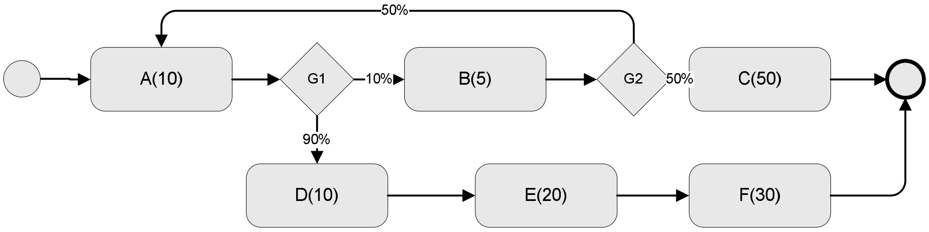


Figure 41: Sequence Flow Loop – Type 2

As can be seen here, when gateway G2 is reached, gateway G1 has already been catered for and added to the stack of probabilities, and it's route to D,E and F may already have been covered in the cost estimate. So, when the sequence flow from G2 is followed, which involves adding in 10% of A and B, its forward probability is not reset to 100% - if it was, this would cause the estimate to be higher. Also, if the sequence flow iteration followed the 90% probability G1 flow (i.e. A,D,E,F), it would never return to G2 – it would 'get lost'. This can be demonstrated by performing the estimate:

- firstly by resetting the alternative sequence flow probability to 100%
- then by leaving it intact
- then by allowing it to 'get lost'

The estimates (resource costs are in brackets in flowchart) would be accumulated as follows:

1. 10 *(A step)*
   0.1 * 5*(B step)*
   0.1 * 0.5 * (10*(A step)* + 5*(B step)*) *– sequence flow calc*
   0.1 * 1 * 50*(C step) – probability reset to 100%*
   0.9 * (10*(D step)* + 20*(E step)* + 30*(F step)*)
   Total = 70.25

2. 10 *(A step)*
   0.1 * 5*(B step)*

0.1 * 0.5 * (10*(A step)* + 5*(B step)*) *– sequence flow calc*

0.1 * 0.5 * 50*(C step) – probability not reset*

0.9 * (10*(D step)* + 20*(E step)* + 30*(F step)*)

Total = 67.75

3. 10 *(A step)*

0.1 * 5*(B step)*

0.1 * 0.5 * 10*(A step) – sequence flow calc*

0.1 * 0.5 * 0.9 * (10*(D step)* + 20*(E step)* + 30*(F step)*) *– incorrect sequence flow*

Total = 13.70

This demonstrates the requirement to deal with a sequence flow which involves other gateways in a different way. This type of gateway is added to the stack of probabilities as its other flow's probability is not reset to 100%.

A real example of this situation is where one decision leads directly to another decision, as in this section of a telephone order process, Figure 42:

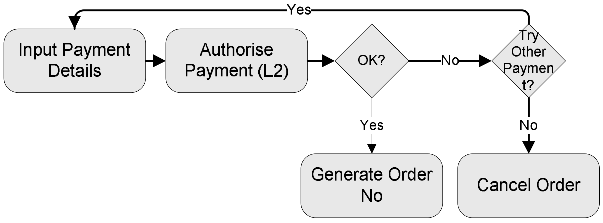


Figure 42: Sequence Flow Loop – Type 2 Example

Another sequence flow loop situation that requires slightly different attention is one where the sequence flows backwards over other gateways, as above, but these gateways' varying flows are effectively completed before the gateway with the sequence flow loop. This generic example, Figure 43, illustrates this situation:

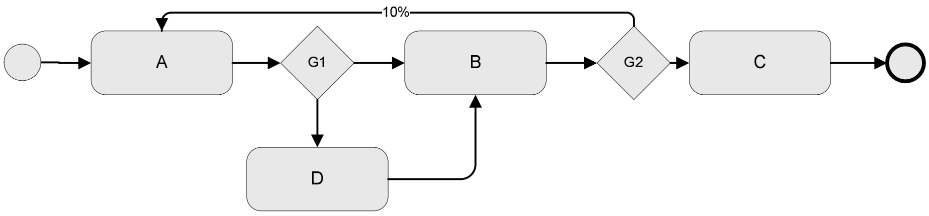


Figure 43: Sequence Flow Loop – Type 3

This situation is treated exactly the same as the one described above, but after the 10% probability of one iteration of the sequence flow loop has been catered for in the cost estimate, the other probability in gateway G2 is reset to 100%.

In the above case and the first case involving sequence flow loops, the gateway with the sequence flow loop is not added to the stack of probabilities – it is immediately navigated through and added into the cost estimate and the other probability reset to 100%, therefore having no impact on previous or later probabilities. The cost estimate then continues using any previous and later probabilities that have been/are accumulated via gateways without sequence flow loops.

6.3.2. Forks and Joins

Another type of gateway which requires catering for is a 'fork' – this is where the process flows outwards on up to 5 different parallel routes. Analysis of a large number of process models suggested that allowing for up to 5 parallel routes was sufficient – however, this could be adjusted if required. All routes are included in the cost estimate with 100% probability. Every fork gateway is ultimately followed by a corresponding 'join' gateway (i.e. 2 flows out indicates 2 flows in) where the parallel flows converge into one. The following generic example, Figure 44, shows a fork and join gateway:

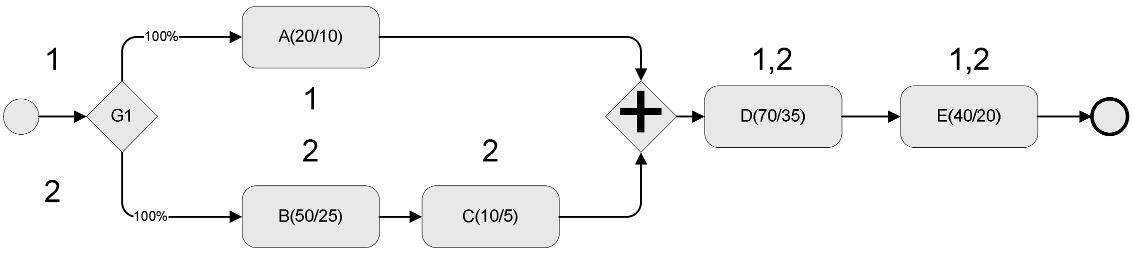


Figure 44: Fork and Join Gateways

As can be seen in the example, gateway G1 forks into 2 parallel paths, both with 100% probability of being executed. They then join together after steps A, B and C have been added into the cost estimate. This, therefore, is a simple estimate with all of the resource costs/durations added into the cost estimate in their entirety (A + B + C + D + E). From an automation/programming perspective, a fork gateway is treated exactly the same as a decision or event based gateway. It is included in the gateway stack of probabilities along with the others but, when the join gateway is encountered, each of the parallel flows' 100% probability is divided by the number of flows and the cost estimate continues. This is done for purposes of efficiency within the technique – it allows for less program code (and less complex code). The alternative would be to return to the fork gateway each time the join gateway is reached and, once its final flow is catered for, continue the estimate - through to the end node in this example. The following table, Table 15, represents the virtual array that is created while the cost estimate for this generic process is taking place:

Table 15: Fork and Join Process – Cost Estimate

PATH	RESOURCE COSTS/ DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/ DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/ DURATION	CUMULATIVE PROBABILITY	TOTAL
1	-	100%	20/10	50%	110/55	50.00%	75.000/37.500
2	-	100%	60/30	50%	110/55	50.00%	115.000/57.500
						100.00%	**190.000/95.000**

The calculation is as follows:

Resource Cost	1 * (20 + (0.5 * 110)) + 1 * (60 + (0.5 * 110)) = 190
Duration	1 * (10 + (0.5 * 55)) + 1 * (30 + (0.5 * 55)) = 95

Which is exactly the same as:

20 + 50 + 10 + 70 + 40 = 190 or 10 + 25 + 5 + 35 + 20 = 95 (i.e. Steps A + B + C + D + E)

6.3.3. ABC Module

The program module which executes this automated cost estimation technique is complex and so a high level pseudo code is used to demonstrate its overall functionality.

All process models presented to the ABC module are feasible (see Chapter 5) – this removes the requirement to check constraints and input/output codes. All processing involves step types, next/previous steps, flows in/out and gateway probabilities. The initial part of the ABC module creates an array and stores only the process details necessary for the cost estimate to take place. Overhead rates for the process, charged by duration, are also stored. The estimate then follows:

```
Start

Get first step after Start Node

Section1: (leads up to first gateway or end if serial process)

Do until not Task step
        Add resource cost to cost total
        Add duration to duration total
        Get next step
Repeat

If End Node – (indicates a purely serial process)
        Cost Estimate = (duration total * overhead rates) + cost total
        Exit module

If not Gateway –
        Get next step
        Go to Section1
```

```
If Gateway with Sequence Flow Loop –
    Record SFL probability
    Retrace steps adding (costs * probability) to cost total and (duration * probability) to_
    duration total
    Note any Gateways crossed
    If Type1 or Type3 SFL –
        Set other Gateway probability to 100%
        Get next step after Gateway
        Go to Section1

Add Gateway details to stack:
    Probability In (Probability Out of previous Gateway or 100)
    Probability Out (Probability In * Gateway probability)
    Index of Gateway probability to follow
    Index of Gateway in array
    Number of flows
Get next step

Section2: (first gateway probability stored – now navigate through all paths accumulating probabilities
          and calculating resource costs and duration)

If Task step –
    Do until not Task step
        Add resource cost * Probability Out to cost total
        Add duration * Probability Out to duration total
        Get next step
    Repeat

If not Gateway or End Node -
    Get next step
    Go to Section2

If Gateway with Sequence Flow Loop –
    Record SFL probability
    Retrace steps adding (costs * probability) to cost total and (duration * probability) to_
    duration total
    Note any Gateways crossed
    If Type1 or Type3 SFL –
        Set other probability to 100%
        Get next step after Gateway
        Go to Section2

If Gateway –
    Add new Gateway details to stack:
        Probability In (Probability Out of previous Gateway)
        Probability Out (Probability In * Gateway probability)
        Index of present Gateway probability to follow
        Index of Gateway in array
        Number of flows

    Modify previous Gateway details:
        Index of Gateway probability to follow
        Number of flows

If End Node
    Do until Number of Flows not = 0
        Return to previous Gateway
        If Number of Flows = 0 –
            If first Gateway –
                Cost Estimate = (duration total * overhead rates) + cost total
                Exit module
        Else
            Modify previous Gateway details:
                Index of Gateway probability to follow
                Number of flows
                Exit Loop
    Repeat

Go to Section2

End
```

The following real-life Telephone Order Process provides a reasonably complex example as it can take eight different routes and contains one sequence flow loop type 1 and one sequence flow loop type 2. It is shown in Figure 45 with costs/durations, probabilities and path numbers:

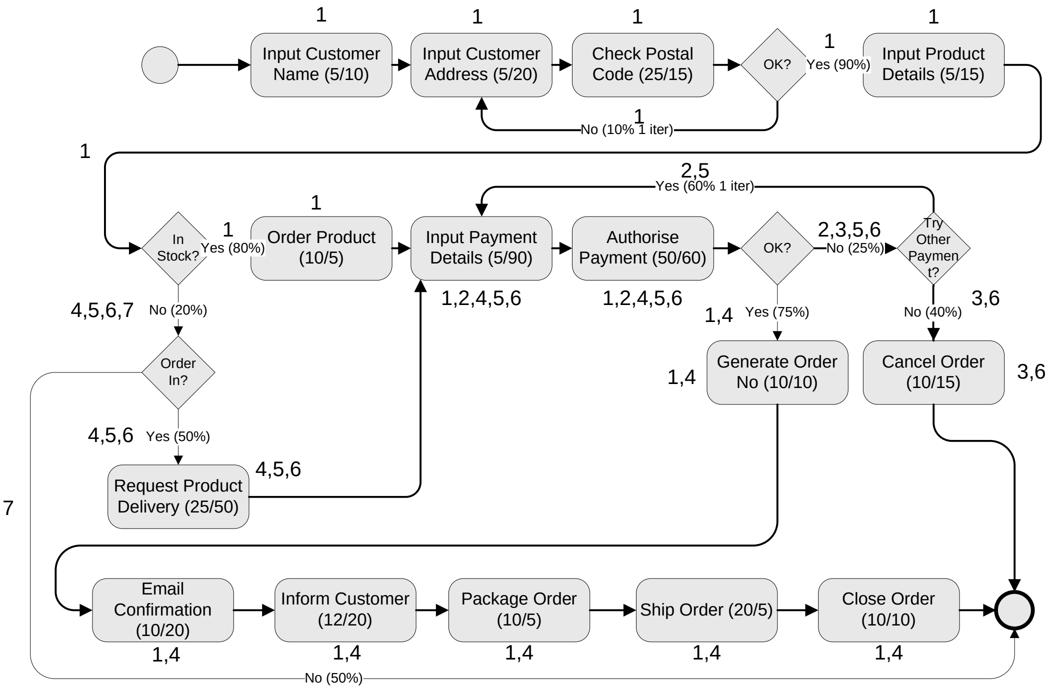


Figure 45: Telephone Order Process

In order to demonstrate the cost estimate, the virtual array that is created during the execution of the technique is shown below in Table 16, as in some of the earlier examples. Also included are the totals for each of the rows that are created. The first gateway with a sequence flow loop is not included in the array as it is dealt with immediately rather than added to the stack of probabilities – however, the second one is included as its other probability is not reset to 100% after the sequence flow loop is catered for. Extra costs generated by the first sequence flow loop are indicated (S) in the array.

Table 16: Telephone Order Process – Cost Estimate

PATH	RESOURCE COSTS/ DURATION	GATEWAY PROBABILITY	GATEWAY PROBABILITY	RESOURCE COSTS/ DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/ DURATION	GATEWAY PROBABILITY	RESOURCE COSTS/ DURATION	CUMULATIVE PROBABILITY	TOTAL
1	35/45+ 3/3.5(S)+ 5/15	80%	-	65/155	75%	72/70	-	-	60%	138.200/ 229.500
2	-	80%	-	-	25%	-	60%	55/150	12%	6.600/ 18.000
3	-	80%	-	-	25%	-	40%	10/15	8%	0.800/ 1.200
4	-	20%	50%	80/200	75%	72/70	-	-	7.50%	13.400/ 25.250
5	-	20%	50%	-	25%	-	60%	55/150	1.50%	0.825/ 2.250
6	-	20%	50%	-	25%	-	40%	10/15	1%	0.100/ 0.150
7	-	20%	50%	-	-	-	-	-	10%	0.000/ 0.000
									100%	**159.925/ 276.35**

To complete the cost estimate, overhead costs are added to the estimate as follows:

Overhead rates = 0.127 per unit of duration (the summation of all four types of overheads, as presented in Figure 36).

Estimated process duration = 276.35 units of duration.

Overheads = 35.096 (i.e. 0.127 * 276.35).

Total cost estimate = 195.021 (i.e. 159.925 + 35.096).

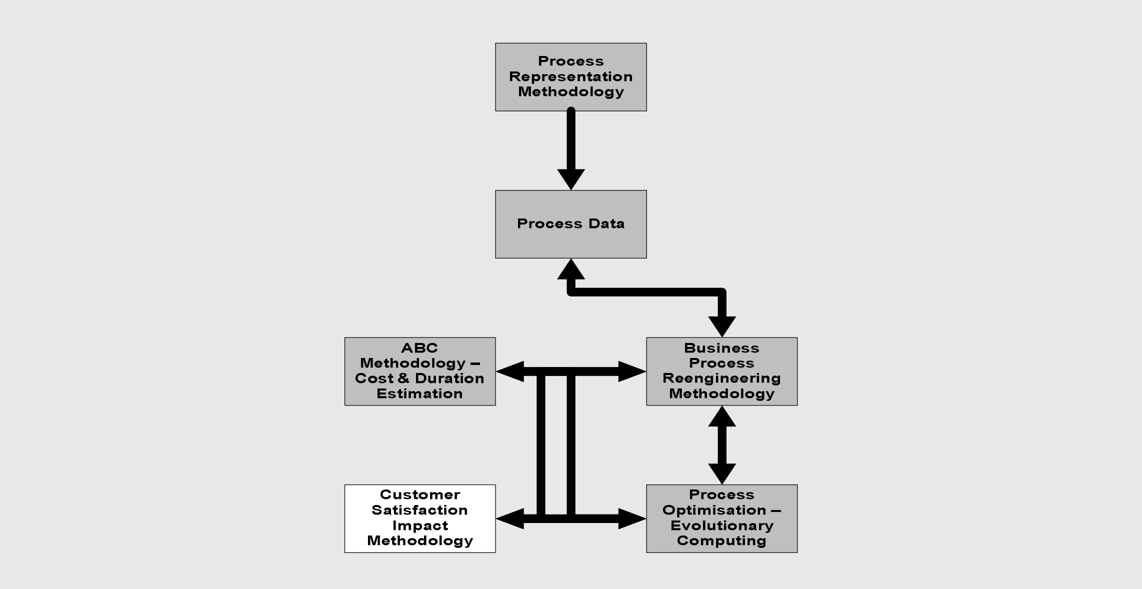
6.4. Validation of ABC Methodology

The responses from the industry survey and ABC forum verified a requirement for the structure and methodology proposed in this research. The accuracy of the methodology is proven by the examples in this chapter, assuming all probabilities, costs and durations are correct – as the cumulative probabilities total 100% in every case, this indicates that all process paths have been catered for in the estimate. Once implemented, the ABC structure and methodology were presented to a number of experts for validation purposes. They included: two industry experts from BT with experience in business process management and business intelligence, one who was already familiar with the project and one who had no prior knowledge of it; a senior marketing analyst specialising in customer churn management; a senior manager with experience of process monitoring; a senior business process analyst specialising in monitoring of customer satisfaction; and a group financial controller specialising in ABC. It was generally well accepted in terms of effectiveness, accuracy and usability, and was perceived as clear and understandable by business and financial systems users. However, it relies heavily on process execution data for duration and probabilities and, in the absence of this data, cannot generate a cost estimate. Therefore, this data must be available, probably via an online transaction processing (OLTP) system and/or workflow logs, and updated regularly in order to maintain accuracy of cost estimates – it was pointed out by the experts that this data may not always be readily available. More details of validation are provided in Chapter 9.

6.5. Summary

The structure and methodology presented in this chapter attempt to add a further level of detail, and therefore accuracy, to an ABC estimate by catering for resources that are not necessarily used each time the process runs – this situation will have been catered for when the process and its activities were originally assigned costs (based on a number of sources, such as past execution data) but, when the process is reengineered, the amount of resource usage may change. This variation occurs because many business processes can follow numerous routes based on the results of events and decisions (Stelling et al, 2008b). In order to maximise the accuracy of the estimate, it would be desirable to store process execution data and periodically update cost, duration and probabilities. The database developed (see Chapter 8) can be used for this purpose. The duration of a process is estimated using the same methodology as the cost estimate, allowing process evaluation based on cost and duration. The next chapter presents the third and final criterion in the evaluation procedure, the impact on levels of customer satisfaction that reengineering a process may generate.

7. Methodology for Measuring Impact on Customer Satisfaction


Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

Estimation of process cost and duration was presented in the previous chapter. As cost and duration data is available, the basis for the estimate is in place. However, this is not the case with levels of customer satisfaction. This area concerns customers' perceptions of the quality of service they considered that they received and whether their expectations were met or exceeded, and this can vary significantly between different customers – what meets one customer's requirements may not meet those of another. The most obvious and commonly used method of measurement is customer feedback – ask the customer for their opinion of the level of service received. Data relating to customer behaviour is also utilised. However, the methodology presented in this chapter requires making an instant judgement on any impact on customer satisfaction that changes to an existing business process could make, and discarding the changes if the impact could be negative - as potentially large numbers of suggested process improvements require evaluation during execution of the technique, the possibility of using customer feedback is not an option. The methodology presented in this chapter attempts to address the research gap identified in Section 2.10: a requirement to develop a methodology which uses available process data to attempt to predict an impact on customer satisfaction.

7.1. Interviews

Two presentations of the research were made to a senior manager working for EDF/Sainsbury's Energy with a background in sales and service process monitoring and tuning, and a senior manager from the industrial sponsors, BT, with a background in process architecture and performance monitoring. The presentation was followed by an interview (See Appendix B for questionnaire) which aimed at answering the overriding question: "How could the effect on customer satisfaction levels that reengineering a process may create be measured without the availability of customer feedback or customer related data". The interviewees shared similar opinions, and the answers and discussion that followed in both cases led to process duration, quality and efficiency. This was also reinforced by the literature review conducted, as presented in Chapter 2.

One of the interviewees named the overriding factors which have an impact on levels of satisfaction as the 'Two Times': 'Cycle Time' and 'Right First Time'.

- Cycle Time – the shorter the duration of the process, the better, in terms of customer satisfaction

- Right First Time – this implies that the business process should execute from beginning to end without problems or failures. As this research deals with the reengineering of business process models, it is assumed that they will not fail. However, with the exception of purely serial processes, processes may follow different routes due to the outcome of events or decisions. In many cases, one route will be less favourable in terms of customer satisfaction than another route – particularly when one of the routes in the model may involve the repetition of a part of the process. Therefore, removal of the possibility of this happening – i.e. removing decision points where possible - can have a positive effect.

Retrospective customer feedback was not an option in the development of the automated technique – this would involve testing potentially very large numbers of varying processes on a group of customers and then collecting their feedback on the service experience. Therefore, the information held in all process models had to be utilised to its maximum effect. This information allowed the ‘Two Times’ to form the basis of the technique.

7.2. Customer Satisfaction Impact Methodology

The two process elements that dictate the cost and duration of a process, as mentioned in the previous chapter, are ‘tasks’ and ‘gateways’. These two elements also are used in the customer satisfaction impact methodology.

Each task has a cost and duration attached to it, and each gateway involves branching in one of a number of directions or branching into parallel paths. The types of gateway that may have an effect on customer satisfaction are decision or event based gateways, where the outcome of an event or decision dictates the direction of the process flow. In this project, each outward flow has a probability attached to it. This is directly relevant to the process cost and duration estimate. As covered in the previous chapter, direct resource costs are totalled according to the percentage probability of them being used, and duration is estimated in a similar manner; the additional/overhead cost rates for the process are then multiplied by the estimated duration to complete the cost estimate (Stelling et al, 2008b). The gateway probabilities are only relevant to customer satisfaction change detection in that one route may be less favourable to the customer than another, but the probabilities are not analysed for this purpose – it is assumed that this may be the case and that there would be less risk of a negative impact on customer satisfaction with as few gateways as possible, particularly when the gateway has a ‘sequence flow loop’ which may involve the repetition of part of the process.

A formula has been developed to calculate a normalised customer satisfaction value for each process generated by the reengineering and optimisation methodology. The development of the formula was an iterative process which needed to consider the following values and combine them in the formula:

- Existing process duration (cycle time)

- Change in process duration
- Number of gateways in existing process (right first time)
- Change in number of gateways

Experimentation led to the inclusion of minimum and maximum possible process durations based on the existing process duration. Therefore, the values used in the formula are:

- Present Process Duration (*pD*)
- Minimum Possible Process Duration (min *D*) – calculated by *pD/10* (this is proposed as an absolute minimum duration and used to estimate a maximum)
- Maximum Possible Process Duration (max *D*) – calculated by *pD* * min *D*
- Alternative Process Duration (*aD*)
- Present Gateway Count (*gC*)
- Alternative Gateway Count (*agC*)

Further fine tuning resulted in the complete formula, as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{pD - aD}{\max D - \min D}\right) * 100 \ + \ \left(\frac{gC - agC}{gC * 2}\right)$$

When this formula generates a negative value, the process being evaluated is essentially rejected – its fitness score is reset to zero. However, when the formula generates a positive value, the value is added to the process's fitness score. This allows for situations where the alternative process cost may be the same or very similar to the existing process, but there appears to be a potential positive effect on customer satisfaction by replacing the existing process with the alternative – as such, the alternative score may be slightly higher due to the addition of the customer satisfaction value. The formula is designed to generate relatively small values (see examples below), so only a very small increase in cost would allow a higher cost alternative process to score more than the existing process due to the addition of the customer satisfaction value, the reasoning being that, although the overriding objective is cost reduction, a small extra process cost may be justified to increase satisfaction levels. For example, if the existing process has an estimated cost of 100 and an alternative process has an estimated cost of 101, via delta comparison the cost score generated will be -1 (i.e. 100-101). Based on cost alone, the alternative will be ranked lower than the existing process and will be unlikely to be selected as a possible replacement. However, if the alternative has generated a value of 2 via the customer satisfaction formula, this is added to the cost score which now becomes +1, thus ranking the alternative as slightly superior to the existing process and making it likely to be selected as a possible replacement – the positive influence on customer satisfaction that this alternative process may have justifies a rise in cost of 1 unit

.

7.2.1. Example One – Cycle Time

The following generic serial process, Figure 46, provides a simple example of the 'cycle time' element of the technique:

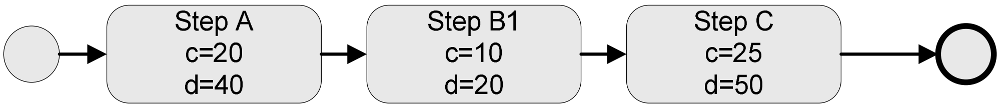


Figure 46: Example 1 – Existing Serial Process

This process has three tasks all with a cost and duration – the cost is 55 units and duration is 110 units. During execution of the reengineering technique, potential replacements are found for Step B1. These are evaluated for cost reduction and then for impact on customer satisfaction. The first is shown below in Figure 47:

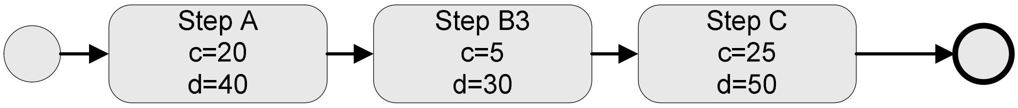


Figure 47: Example 1 – First Alternative Process

Step B3 has a lower cost which reduces the process cost to 50 units. This makes it an acceptable replacement for Step B1. However, it has a duration of 30 units as opposed to 20, which increases the duration of the process to 120. min *D* is calculated as 11 (110/10) and max *D* as 1210 (11*110). The calculation to generate the customer satisfaction score is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{110-120}{1210-11}\right)*100 + \left(\frac{0-0}{0*2}\right) = -0.834$$

The small rise in process duration creates a customer satisfaction ‘score’ of less than zero. Therefore, although this suggested process has a lower cost, it is rejected due to the risk of a negative effect on satisfaction. Another alternative is then evaluated, as shown in Figure 48:

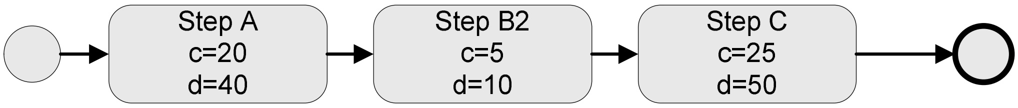


Figure 48: Example 1 – Second Alternative Process

Step B2, as in the example above, has a lower cost which reduces the process cost to 50 units. This makes it an acceptable replacement for Step B1. It also has a duration of 10 units as opposed to 20, which reduces the duration of the process to 100. The calculation to determine the effect on customer satisfaction is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{110-100}{1210-11}\right)*100 + \left(\frac{0-0}{0*2}\right) = +0.834$$

The small fall in process duration creates a customer satisfaction ‘score’ of more than zero. Therefore, this suggested process has a lower cost and may also a positive impact on customer satisfaction – it would be accepted as a replacement for the existing process for these reasons.

7.2.2. Example Two – Right First Time

The following generic process, Figure 49, this time containing a gateway, provides a simple example of the 'right first time' element of the technique:

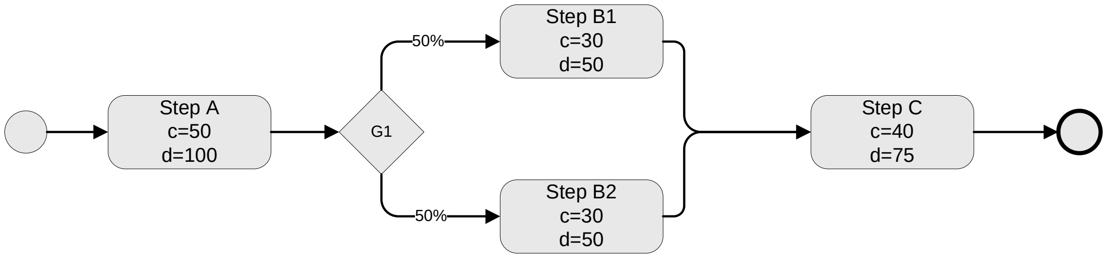


Figure 49: Example 2 – Existing Process with Gateway

This process has four tasks and one gateway, Steps B1 and B2 having a 50% chance each of being executed. The cost estimate is therefore 120 units and duration estimate is 225 units. During execution of the reengineering technique, potential replacements are found for Gateway G1 and steps B1 and B2. These are evaluated for cost reduction and then for impact on customer satisfaction. The first is shown below in Figure 50:

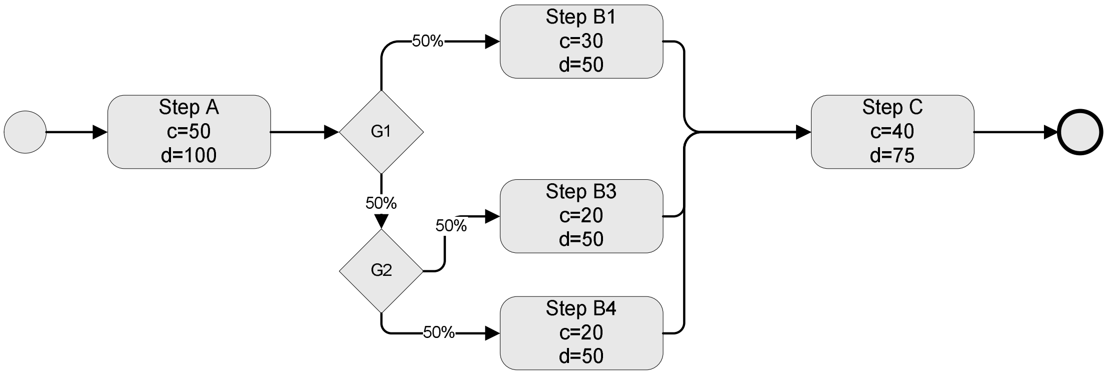


Figure 50: Example 2 – First Alternative Process

The duration estimate of this potential process is exactly the same as the existing process but the cost estimate is reduced to 115. However, it now has an extra gateway. min *D* is calculated as 22.5 (225/10) and max *D* as 5062.5 (22.5*225). The calculation to determine the effect on customer satisfaction is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{225-225}{5062.5-22.5}\right)*100 + \left(\frac{1-2}{1*2}\right) = -0.5$$

When this process is evaluated for impact on customer satisfaction, a negative score is generated due to the addition of a gateway – there is a risk that this gateway will have an adverse effect. Another alternative is then evaluated, as in Figure 51:

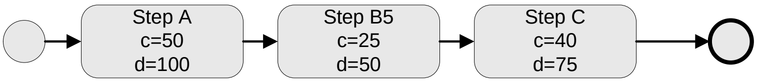


Figure 51: Example 2 – Second Alternative Process

This process again has the same duration as the existing process with a lower cost of 115, but now the gateway has been removed. The calculation is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{225-225}{5062.5-22.5}\right)*100 + \left(\frac{1-0}{1*2}\right) = +0.5$$

Therefore, this suggested process change has a positive impact on customer satisfaction due to the removal of the gateway and its associated negative impact on customer satisfaction.

7.2.3. Example Three – The Two Times

The following example, Figure 52, this time demonstrates both the cycle time and right first time included in the evaluation of a generic process.

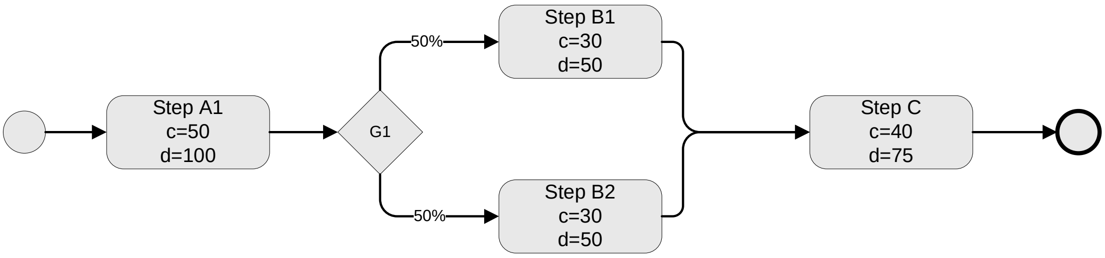


Figure 52: Example 3 – Existing Process with Gateway

This process has four tasks and one gateway, Steps B1 and B2 having a 50% chance each of being executed. The cost estimate is therefore 120 units and duration estimate is 225 units. During execution of the reengineering technique, potential replacements are found for Step A1, Gateway G1 and steps B1 and B2. These are evaluated for cost reduction and then for impact on customer satisfaction. The first is shown below in Figure 53:

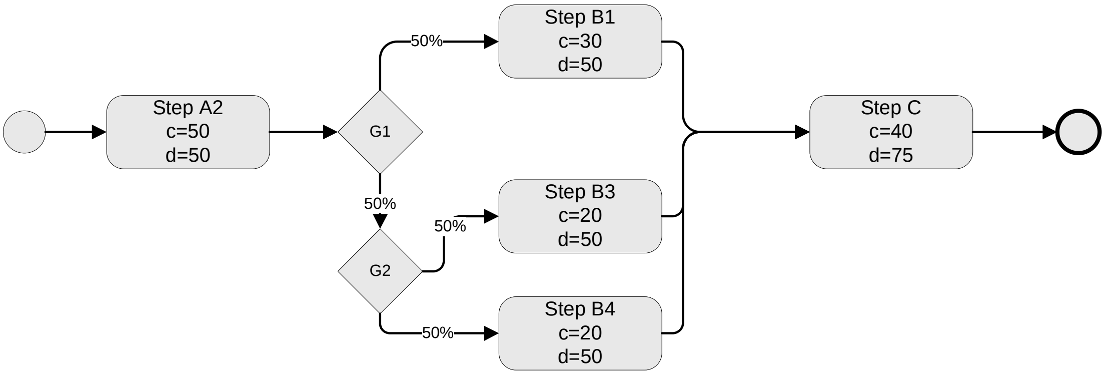


Figure 53: Example 3 – First Alternative Process

The duration estimate of this potential process is reduced by 50 due to the replacement of Step A1 with Step A2, and the cost estimate is reduced to 115. However, it now has an extra gateway. min *D* is calculated as 22.5 (225/10) and max *D* as 5062.5 (22.5*225). The calculation to determine the effect on customer satisfaction is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{225-175}{5062.5-22.5}\right)*100 + \left(\frac{1-2}{1*2}\right) = +0.492$$

When this process is evaluated for impact on customer satisfaction, a positive score is generated due to the large reduction in duration, but the score is reduced to approximately half due to the addition of a gateway. If the duration reduction had been 25, for example, a negative score would have been generated. Another alternative, Figure 54, is then evaluated:

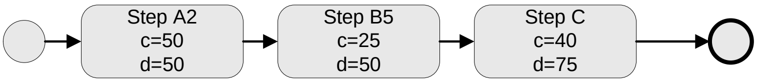


Figure 54: Example 3 – Second Alternative Process

The duration estimate of this potential process is reduced by 50 due to the replacement of Step A1 with Step A2, and the cost estimate is reduced to 115 due to the removal of gateway G1 and replacement of Steps B1 and B2 with B5. The calculation is as follows:

$$\text{Customer Satisfaction Value} = \left(\frac{225-175}{5062.5-22.5}\right)*100 + \left(\frac{1-0}{1*2}\right) = +1.492$$

Due to the large duration reduction and the removal of a gateway, this process generates a score which would make it a favourable replacement for the existing process with a good chance of positively impacting customer satisfaction.

7.3. Validation of Customer Satisfaction Methodology

Following implementation, the methodology was validated using test scenarios, presented in Chapter 9, which demonstrate the appropriate rises and falls in customer satisfaction values based on the Two Times criteria. Expert opinion was also employed: the methodology was presented to six experts who were then asked to provide ratings of effectiveness, accuracy and usability, or level of agreement with a statement (i.e. disagree, agree), plus comments, on the following aspects:

- ‘Cycle Time’ and ‘Right First Time’ criteria.
- Utilisation of all information held in process models to achieve indication of negative or positive effect
- This is potentially the only automated means of providing an indication without feedback from customer or customer related data
- The customer satisfaction formula
- Use of value produced by formula (e.g. slight edge in selection for process with positive; reject negative)

Full details of test scenarios and expert opinion are presented in Chapter 9.

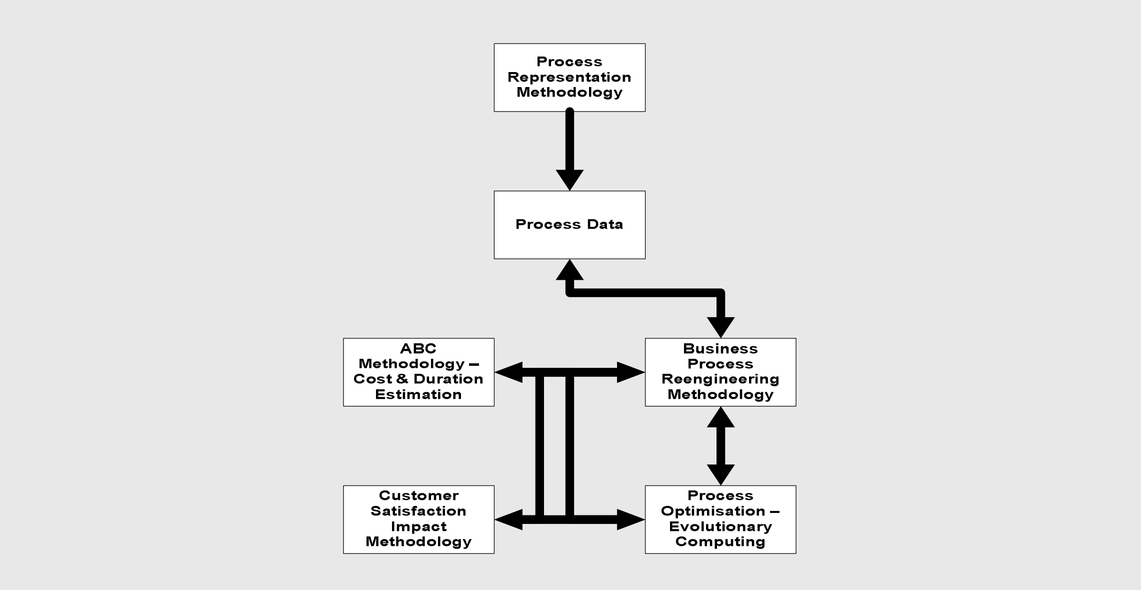
7.4. Summary

The methodology presented in this chapter does not attempt to measure customer satisfaction – it attempts to provide an indication of the potential impact, positive as well as negative, that reengineering a service based process may create. In order to achieve this, a small, normalised value is generated via a formula which utilises delta comparisons between existing and alternative process duration, and number of decision points. A negative value indicates a risk of negative impact, and so the suggested process is rejected; a positive value indicates a possible positive impact, and so the process' suitability for selection as a replacement is weighted accordingly. This formula can provide a reasonably accurate means of predicting impact of process reengineering. However, although the majority of customers appreciate a swift, efficient and uncomplicated process, there will always be a small minority who may appreciate spending a longer time on the phone or the internet – reducing cycle time may have a negative impact for these customers. Furthermore, a reduction in the duration of a process may detract from its effectiveness, thus having a negative impact. Also, the removal of gateways may not necessarily have a positive impact on satisfaction levels: some gateways will allow a customer to quickly talk to the correct people or arrive at the correct web page, and each flow out of the gateway will be equally beneficial - removal of this type of gateway may risk creating a negative impact. In general though, the technique provides as accurate an indication as possible with the information available within the elements of a business process model, in a situation where retrospective feedback is not an available option (Stelling et al, 2008c).

The ideas behind the methodology are based on literature review and expert opinion. Examples were given, both generic and real-life, to demonstrate how two criteria, 'cycle time' and 'right first time', are used to generate an indication of impact on customer satisfaction that process reengineering may create.

The following chapter describes the integration and implementation of the techniques detailed in Chapters 4, 5, 6 and 7.

8. Integration and Implementation of Automated Business Process Reengineering Approach


Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

The previous four chapters have presented the components of the automated reengineering approach. These are illustrated in Figure 55.

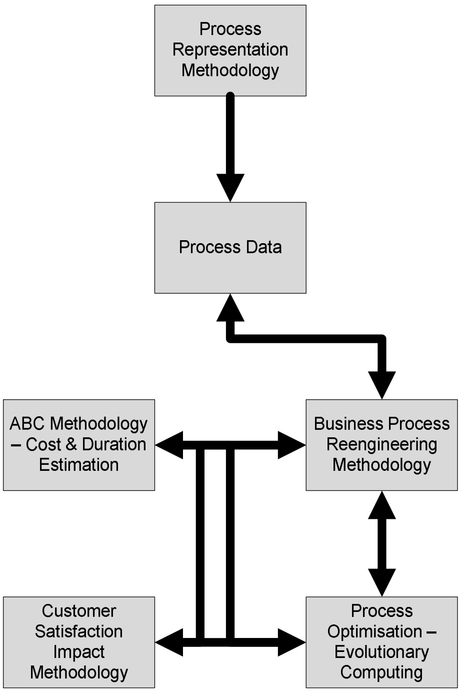

Process Representation Methodology
Process Data
ABC Methodology – Cost & Duration Estimation
Business Process Reengineering Methodology
Customer Satisfaction Impact Methodology
Process Optimisation – Evolutionary Computing

Figure 55: Components of Automated Reengineering Approach

Chapter four presented the proposed process coding structure and mechanism; Chapter five presented the automated stages of process reengineering and the optimisation module; Chapter six described the ABC methodology which generates a process cost and duration estimate; and Chapter seven detailed the methodology for detecting and measuring impact on levels of customer satisfaction. This chapter presents the integration of these techniques, as illustrated above in Figure 55, via a prototype software platform: an overview of the users, the user interface, and inputs to and outputs from the platform is provided; the design and development of the relational database is described, with examples of some of the tables; the objectives and constraints based on the values generated by the component techniques, as presented in Chapters 6 and 7, are analysed in combination, and a demonstration of how they are used to allocate a fitness score given; and a sample 'tour' of the system, ABPRS, is provided.

8.1. Integration via Prototype Software Platform

In order to integrate the methodologies described in Chapters 4, 5, 6 and 7, a prototype software platform was developed. This section provides an overview of the platform.

8.1.1. Users

The principal users of the system will be management with an interest in process monitoring. These could, for example, be process architects and process owners.

8.1.2. User Interface

There is a basic user interface which allows the user to:

- View/update all available process data including existing process models.
- View/update cost data (resource centres with resources and costs, activity centres with resources and overhead costs).
- View/update process duration data.
- Select processes for reengineering.
- Execute the reengineering algorithm.

- View output from execution of the algorithm
- If satisfied with output, reengineer the selected process

8.1.3. Inputs

- Selected coded process model
- All available process data (process models, individual process steps, sub processes)
- Cost and duration data

8.1.4. Outputs

- Report detailing optimum process (which should contain all details necessary to allow translation into flow diagram)
- Chart detailing all changes in process cost from existing process
- Subject to acceptance of changes, a reengineered process

8.1.5. Database Development

A period of experimentation was undertaken in the early stages of the research using MATLAB – this facilitated the initial development of the coding mechanism (Chapter 4). However, MATLAB lacks a relational database facility and so text files were used as tables for basic testing. Therefore, a self-contained database package was required. As Microsoft Access is a relatively user friendly database which is familiar to many people and can be found in most organisations, and also provides a Visual Basic (VB) Project facility in which to develop program code and a Structured Query Language (SQL) builder, this was chosen as the tool for further development.

All process related data is held in the database. The database schema reflects the taxonomy structure and coding mechanism described in Chapter 4, and also reflects the cost estimating structure described in Chapter 6. As reviewed in Chapter 2, an ER modelling approach was adopted for the database design. All 19 tables (or entities) and relationships, plus connectivity and existence where a relationship exists (some tables are free-standing), are summarised in Table 17:

Table 17: Entity Relationship Summary

Entity/Table	Connectivity	To	Key	Existence
Domain L1	one to many	Domain L2	Domain L1 Code	Mandatory
Domain L2	one to many	Domain L3	Domain L2 Code	Mandatory
Domain L3	one to many	Activity	Domain L3 Code	Mandatory
Activity	one to many	Transaction	Activity Code	Mandatory
	many to one	Activity Centre	Activity Centre Code	Mandatory
Transaction	one to many	Sub-Transaction	Transaction Code	Mandatory
Sub-Transaction	one to one	Resource Costs	Resource Code	Optional
	one to one	Gateway Probabilities	Domain L1 Code Domain L2 Code Domain L3 Code Activity Code Transaction Code Sub-Transaction Code = Unique 10 Digit ID	Optional
Activity Centre	one to many	Activity	Row ID	Mandatory
	one to one	Overhead Costs	Row ID	Mandatory
	one to many	Resource Costs	Row ID	Mandatory
Resource Costs	one to one	Sub-Transaction	Resource Code	Optional
	many to one	Resource Centre	Resource Centre Code	Mandatory
	many to one	Activity Centre	Activity Centre Code	Mandatory
Resource Centre	one to many	Resource Costs	Row ID	Mandatory
Overhead Costs	one to one	Activity Centre	Row ID	Mandatory
Gateway Probabilities	one to one	Sub-Transaction	Unique 10 Digit ID	Optional
Constraints	none			
Inputs	none			
Outputs	none			
Flows	none			
Step Types	none			
Process Sequences	none			
Sub Process Library	none			
(Process Execution Log)	none			

The attributes of the tables, notably Sub-Transaction (i.e. process step), Process Sequences and Sub Process Library, are mainly the process representation codes described in Chapter 4, along with cost, duration and probabilities, held in tables Overhead Costs, Resource Costs and Gateway Probabilities. Other tables are look-up tables used for decoding purposes when displaying data to the user. Process Execution Log is not physically present in the database but is an addition that has been catered for and can be introduced by an organisation when process execution data becomes available. The database schema has been modelled using Microsoft Visio, as shown in Figure 56, and is formatted to reflect the taxonomy/coding mechanism and cost estimating structure:

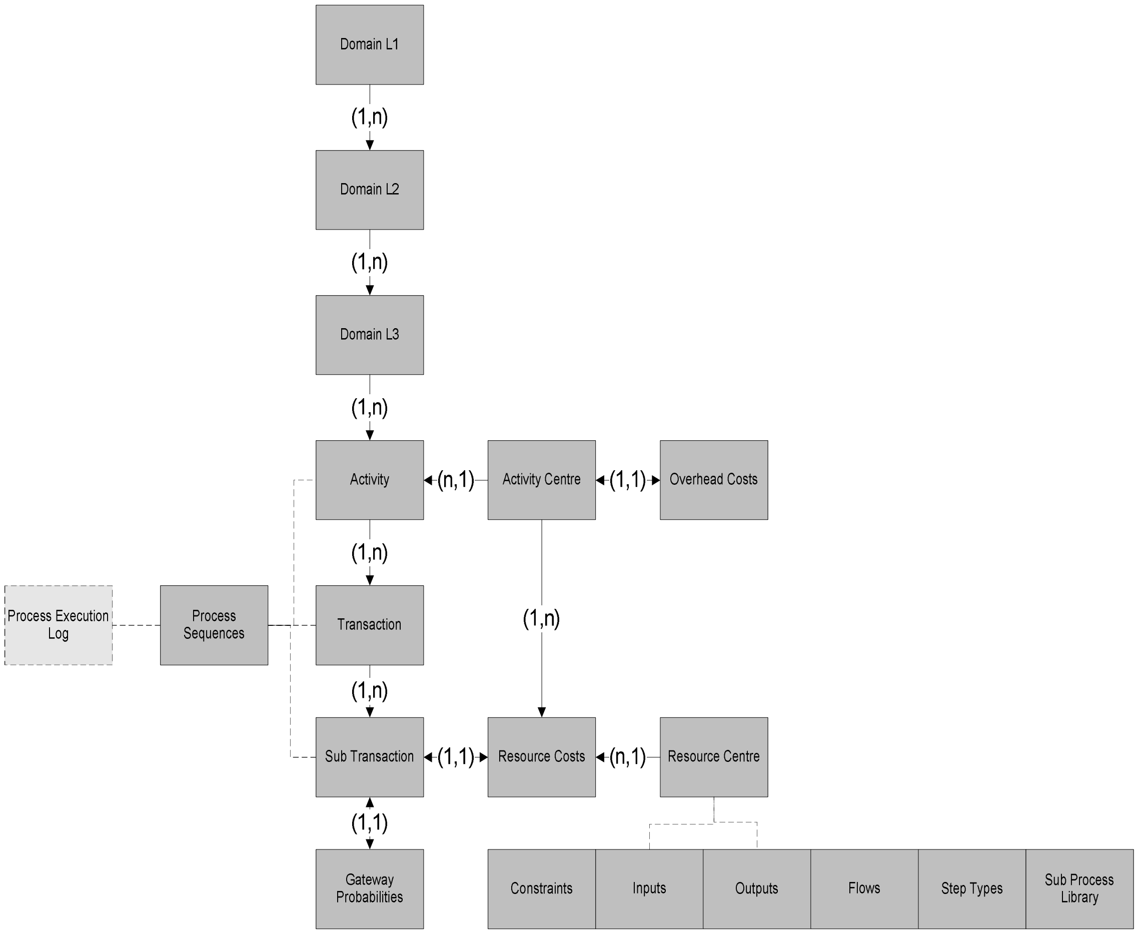
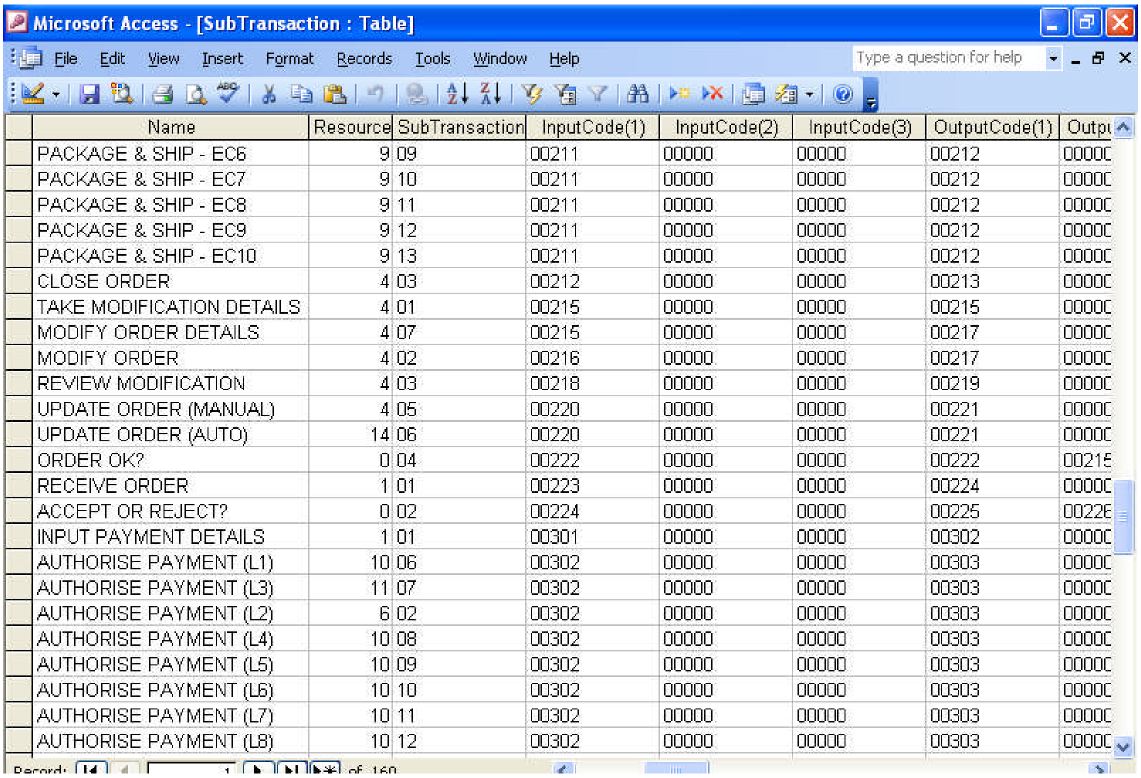


Figure 56: Database Schema Diagram

The schema diagram indicates the entity relationships specified in Table 17. Dashed lines imply a connection between tables, but without a set relationship. A complete list of tables and fields is presented in Appendix D. An example snapshot of table SubTransaction, the table which contains all available process steps, is shown below in Figure 57.



Name	Resource	SubTransaction	InputCode(1)	InputCode(2)	InputCode(3)	OutputCode(1)	Outpu
PACKAGE & SHIP - EC6	9	09	00211	00000	00000	00212	0000C
PACKAGE & SHIP - EC7	9	10	00211	00000	00000	00212	0000C
PACKAGE & SHIP - EC8	9	11	00211	00000	00000	00212	0000C
PACKAGE & SHIP - EC9	9	12	00211	00000	00000	00212	0000C
PACKAGE & SHIP - EC10	9	13	00211	00000	00000	00212	0000C
CLOSE ORDER	4	03	00212	00000	00000	00213	0000C
TAKE MODIFICATION DETAILS	4	01	00215	00000	00000	00215	0000C
MODIFY ORDER DETAILS	4	07	00215	00000	00000	00217	0000C
MODIFY ORDER	4	02	00216	00000	00000	00217	0000C
REVIEW MODIFICATION	4	03	00218	00000	00000	00219	0000C
UPDATE ORDER (MANUAL)	4	05	00220	00000	00000	00221	0000C
UPDATE ORDER (AUTO)	14	06	00220	00000	00000	00221	0000C
ORDER OK?	0	04	00222	00000	00000	00222	00215
RECEIVE ORDER	1	01	00223	00000	00000	00224	0000C
ACCEPT OR REJECT?	0	02	00224	00000	00000	00225	0022E
INPUT PAYMENT DETAILS	1	01	00301	00000	00000	00302	0000C
AUTHORISE PAYMENT (L1)	10	06	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L3)	11	07	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L2)	6	02	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L4)	10	08	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L5)	10	09	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L6)	10	10	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L7)	10	11	00302	00000	00000	00303	0000C
AUTHORISE PAYMENT (L8)	10	12	00302	00000	00000	00303	0000C

Figure 57: Example Process Data

8.1.6. Functionality

The principal functionality of the software platform is the reengineering algorithm, the details of which were presented in Chapter 5, and which is illustrated in the following program structure chart, Figure 58:

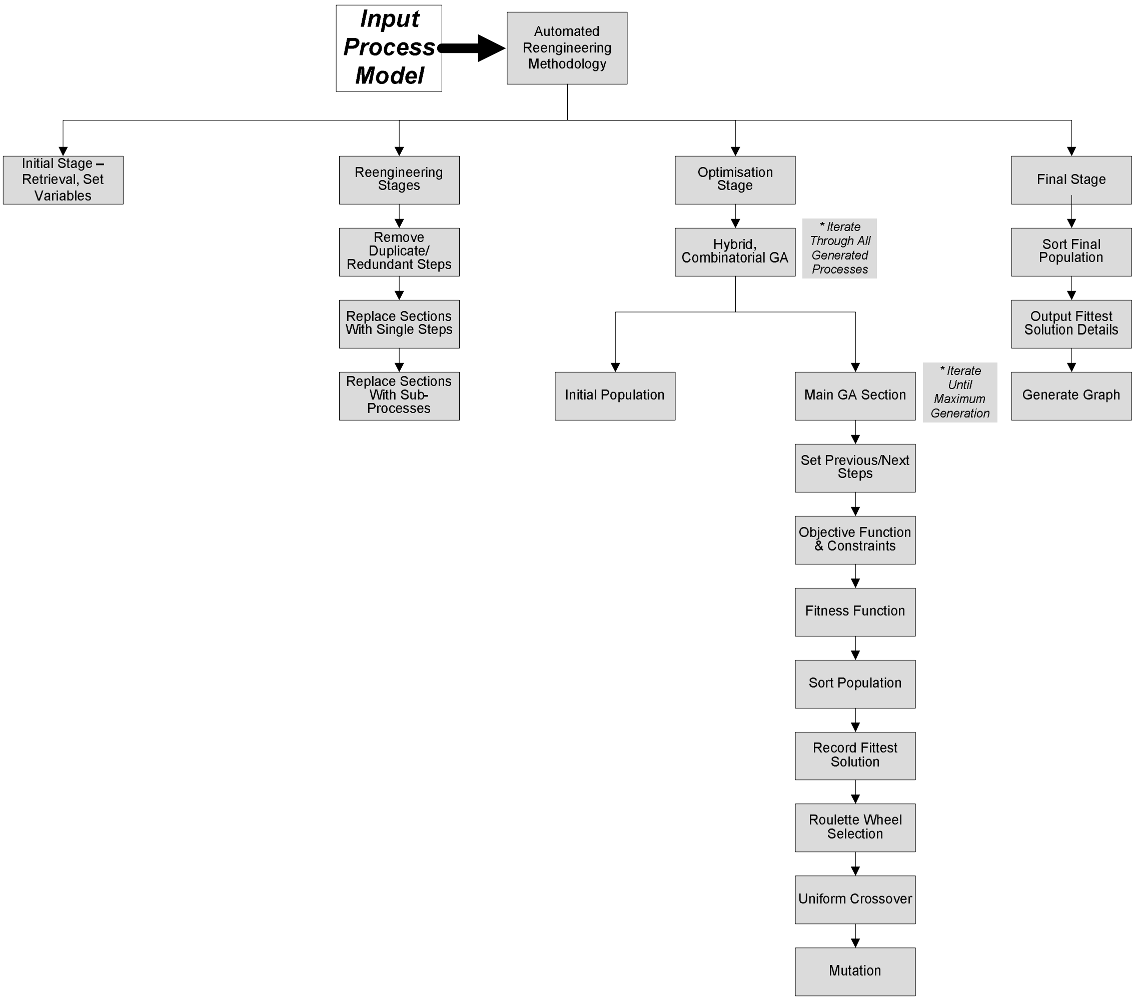

Input Process Model
Automated Reengineering Methodology
Initial Stage – Retrieval, Set Variables
Reengineering Stages
Remove Duplicate/ Redundant Steps
Replace Sections With Single Steps
Replace Sections With Sub-Processes
Optimisation Stage
Hybrid, Combinatorial GA
* Iterate Through All Generated Processes
Initial Population
Main GA Section
* Iterate Until Maximum Generation
Set Previous/Next Steps
Objective Function & Constraints
Fitness Function
Sort Population
Record Fittest Solution
Roulette Wheel Selection
Uniform Crossover
Mutation
Final Stage
Sort Final Population
Output Fittest Solution Details
Generate Graph

Figure 58: Modular Structure of ABPRS

Each part of the above chart is presented in detail in Chapter 5 with pseudo code and examples. The method for allocating a fitness to processes, the 'Fitness Function' module in the above chart, has been mentioned in Chapters 5, 6 and 7, and is now described fully.

Reengineering Objectives and Constraints

Evaluation of processes is achieved using the methodologies presented in Chapters 6 and 7. The three values generated by the reengineering system, cost, duration and impact on customer satisfaction, are now described in combination - how they are linked, and how they are used to allocate a level of fitness to processes.

In the initial stages of the research, it appeared that the three values indicated a multi-objective optimisation problem which may have utilised available algorithms in the optimisation module, such as NSGAII and SPEA. However, as mentioned in Chapter 2, these types of algorithms deal with independent or non-dominated objective values in which the improvement of one value has an adverse effect on another value. As a result, they generate a set of compromise solutions – a set of pareto-optimal points (or a pareto front). It became apparent as the ABC and customer satisfaction methodologies were developed, that the values in this research are not independent or non-dominated for the following reasons:

- *Duration* - estimated using the same method as cost estimation. It is multiplied by overhead costs and added to resource cost estimate. It is also used as one of the two criteria for indicating changes in customer satisfaction

  Therefore:

  - If duration decreases, cost decreases, customer satisfaction increases
  - If duration increases, cost increases, customer satisfaction decreases

- *Cost* - is estimated via an ABC estimation module in which resource costs are totalled based on probabilities of the process using them. Overhead cost rates per activity are multiplied by the estimated process duration and added to the total estimated resource costs of the process.

  Therefore:

  - If cost increases, it is possible that duration will have increased and so customer satisfaction will have decreased
  - If cost decreases, it is possible that duration will have decreased and so customer satisfaction will have increased

- *Customer Satisfaction* - is calculated using estimated process duration and removal of decision points (or Gateways) from a process.

Therefore:

- If customer satisfaction increases, it is likely that duration will have decreased and so cost will have decreased - this is because removal of decision points indicates removal of at least one constrained task step which provides the input to the decision, and so duration and cost have been reduced by this, or duration only has decreased, or both of these situations have occurred.

- If customer satisfaction decreases, it is likely that duration will have increased and so cost will have increased - this is because addition of decision points indicates addition of at least one constrained task step which provides the input to the decision, and so duration and cost have been increased by this, or duration only has increased, or both of these situations have occurred.

This relationship between the values can be illustrated as follows in Figure 59:

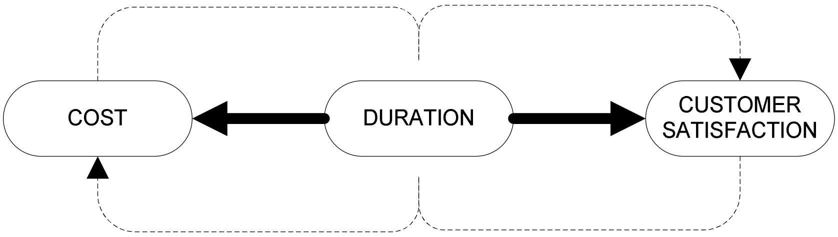


Figure 59: Objective Value Relationships

There is a direct relationship between duration and cost, and duration and customer satisfaction, and an indirect relationship between cost and customer satisfaction. Therefore, these relationships between generated values supported a single objective situation in which process cost is the objective, but with constraints incorporated that also take into consideration process duration and customer satisfaction levels as follows:

- Excessive process duration is calculated as
  Present Duration * (Present Duration / 10).

  This calculation provides a reasonable threshold to check reengineered processes against. If the reengineered process duration exceeds this value, its fitness score is set to zero.

- As described in the previous chapter, if the customer satisfaction impact formula generates a negative value, the fitness score is set to zero; a positive value is added to the fitness score to give the process an edge in future selection over processes of similar cost but lacking positive impact on satisfaction.

Fitness Function

The module which uses the cost, duration and customer satisfaction values to allocate a score to each process generated can be represented in pseudo code as follows:

```
Minimum Process Duration = Existing Process Duration / 10
Maximum Process Duration = Existing Process Duration * Minimum Process Duration

If Estimated Process Duration > Maximum Process Duration Or
Customer Satisfaction Value < 0 Then
   Fitness Score = 0
Else
   Fitness Score = (Existing Process Cost – Estimated Process Cost) + Customer Satisfaction Value
End If
```

Each generated process is allocated a fitness, and then all processes are sorted. Full details are presented in Chapter 5.

8.2. Tour of Prototype Software Platform: ABPRS

The prototype software platform, ABPRS, has a user interface which provides the facility to inspect process, cost and duration data, and the ability to select processes and execute the automated reengineering algorithm. The bulk of the platform’s functionality is the VB code, of which there are 2,314 lines, along with numerous SQL queries for purposes of data retrieval and storage. The architecture of ABPRS is illustrated in Figure 60:

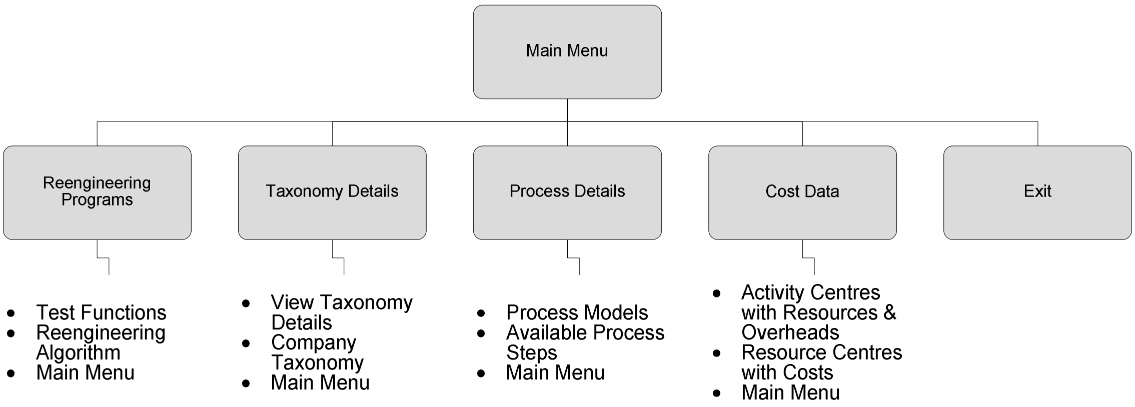


Figure 60: ABPRS Architecture

8.2.1. Main Menu

The main menu is illustrated in Figure 61:

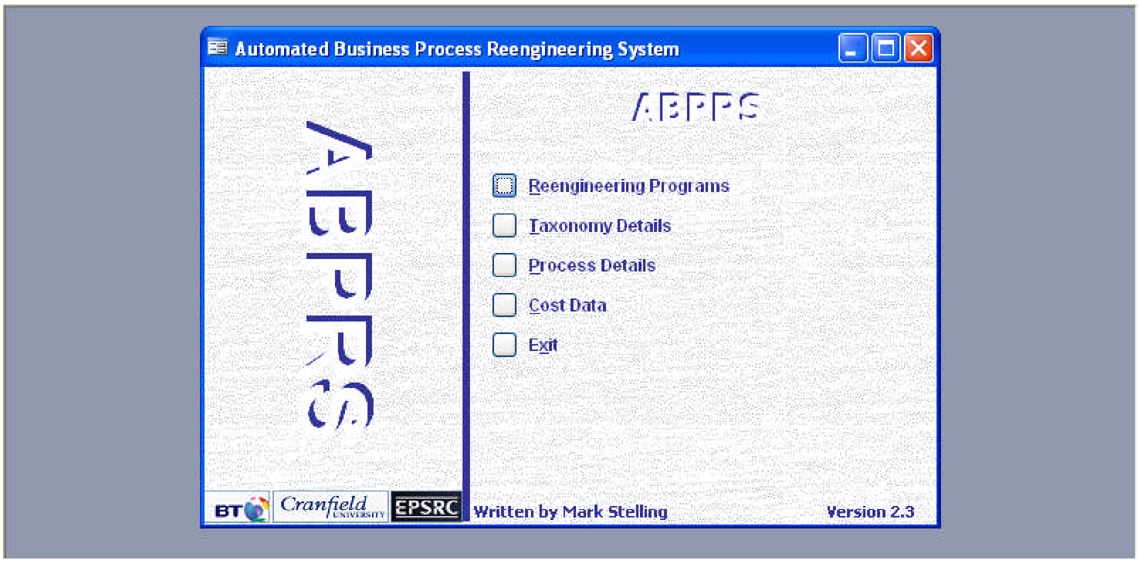


Figure 61: ABPRS - Main Menu

There are four sub-menus: Reengineering programs, Taxonomy Details, Process Details and Cost Data. A facility to exit the application is also provided.

8.2.2. Reengineering Programs

The sub-menu, Reengineering Programs, is shown below in Figure 62:

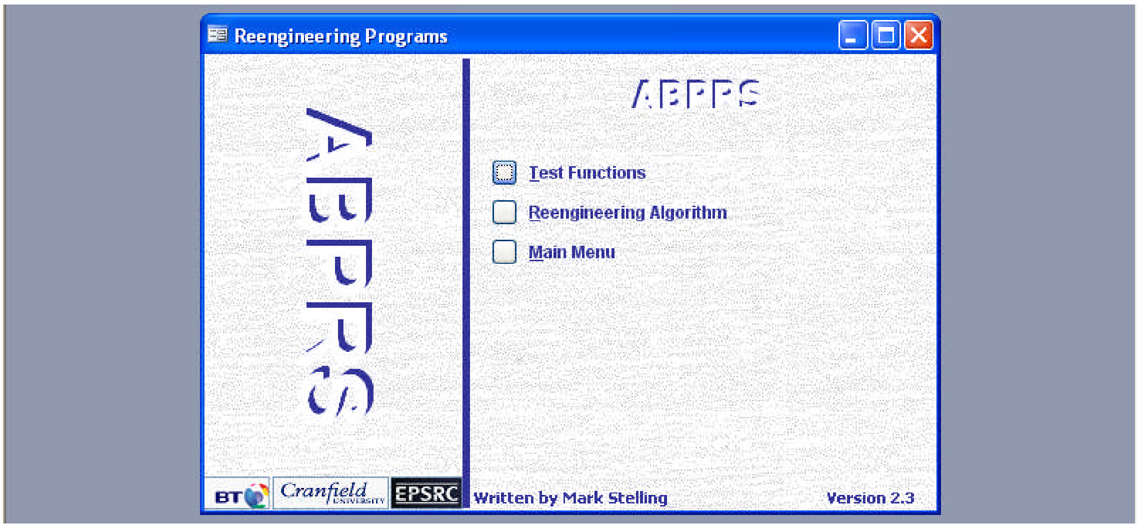
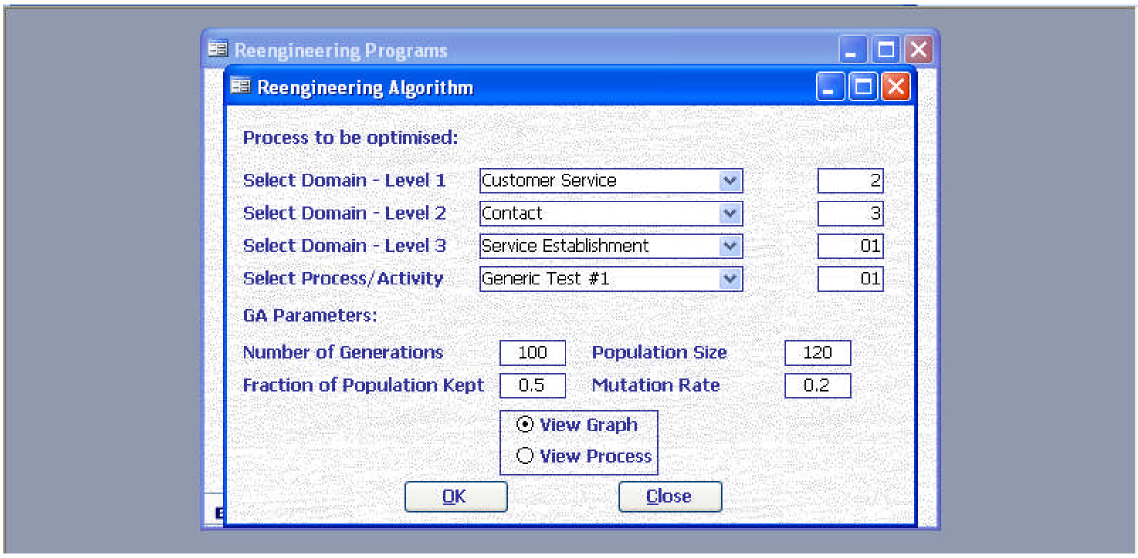


Figure 62: ABPRS - Reengineering Programs Menu

This provides the facility to run a test function, initially incorporated into the system to test the functionality of the GA based module. It also allows selection of processes for reengineering via the ‘Reengineering Algorithm’ option, as follows in Figure 63:



Process to be optimised:

Select Domain - Level 1	Customer Service	2
Select Domain - Level 2	Contact	3
Select Domain - Level 3	Service Establishment	01
Select Process/Activity	Generic Test #1	01

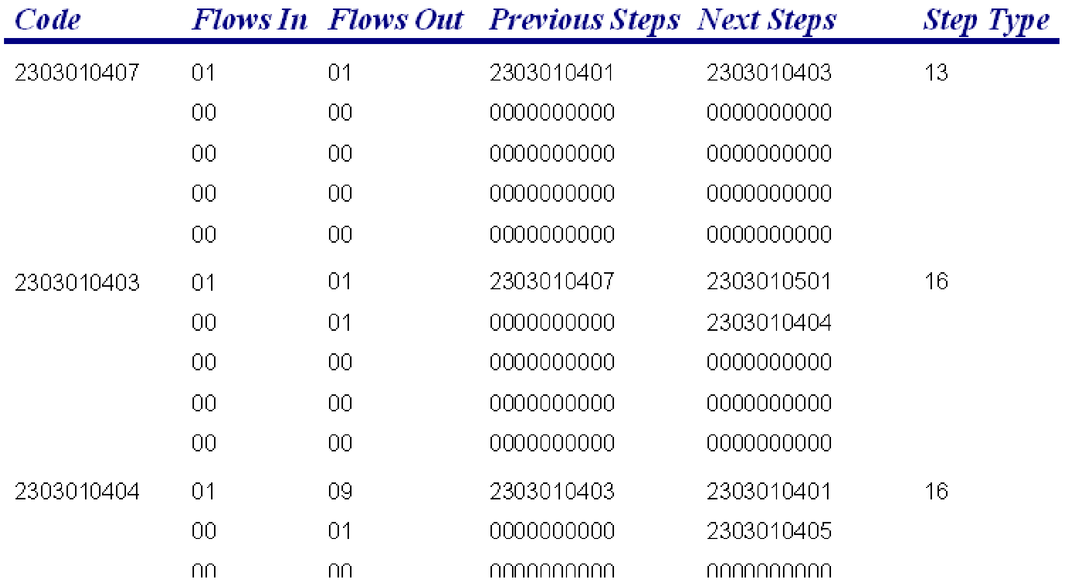
GA Parameters:

Number of Generations	100	Population Size	120
Fraction of Population Kept	0.5	Mutation Rate	0.2

● View Graph
○ View Process

Figure 63: ABPRS - Reengineering Algorithm

This form allows the user to select the required process for reengineering and to supply parameters for the GA based optimisation module (although default values for this are provided). After execution of the algorithm, a report is generated, detailing the optimum process found by the algorithm – a snapshot of a report is shown in Figure 64:



Code	Flows In	Flows Out	Previous Steps	Next Steps	Step Type
2303010407	01	01	2303010401	2303010403	13
	00	00	0000000000	0000000000	
	00	00	0000000000	0000000000	
	00	00	0000000000	0000000000	
	00	00	0000000000	0000000000	
2303010403	01	01	2303010407	2303010501	16
	00	01	0000000000	2303010404	
	00	00	0000000000	0000000000	
	00	00	0000000000	0000000000	
	00	00	0000000000	0000000000	
2303010404	01	09	2303010403	2303010401	16
	00	01	0000000000	2303010405	
	00	00	0000000000	0000000000	

Figure 64: ABPRS – Reengineered Process Report

The details provided by the report allow its simple translation into a flow diagram. An example of a chart produced by the reengineering algorithm, linked to MS Excel, is shown in Figure 65:

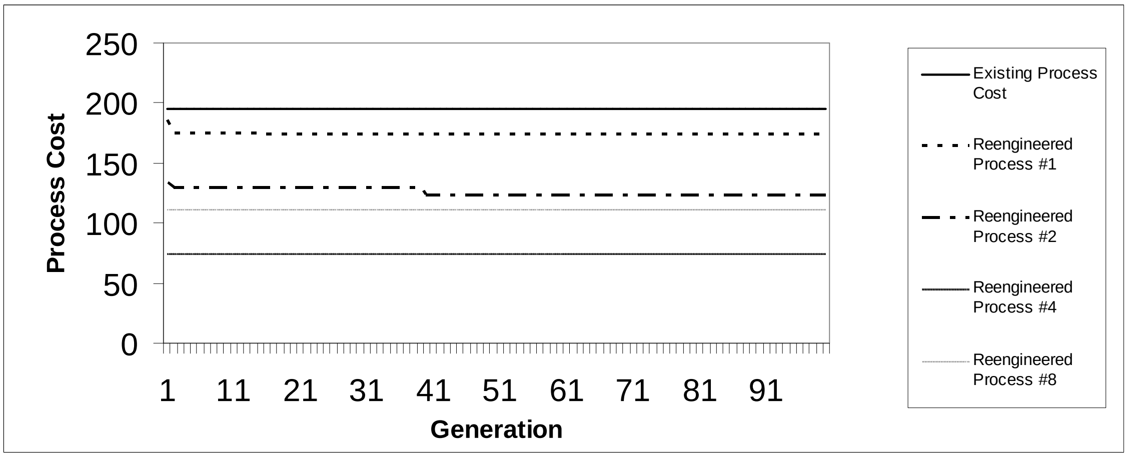


Figure 65: ABPRS – Reengineered Process Chart

The chart details changes in cost for each unique process output from the reengineering stages of the algorithm into the optimisation stage.

8.2.3. Taxonomy Details

The sub-menu, Taxonomy Details, is shown below in Figure 66:

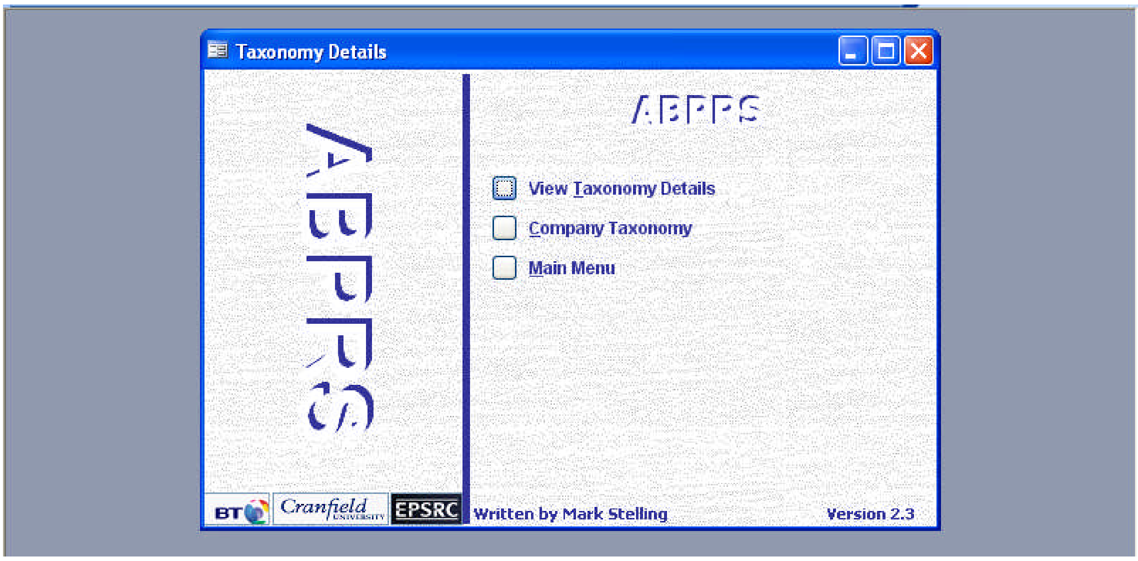


Figure 66: ABPRS - Taxonomy Details Menu

This menu allows the user to examine the structure of the company taxonomy, one level at a time or as a diagram depicting the structure, as shown in Figures 67 and 68 below:

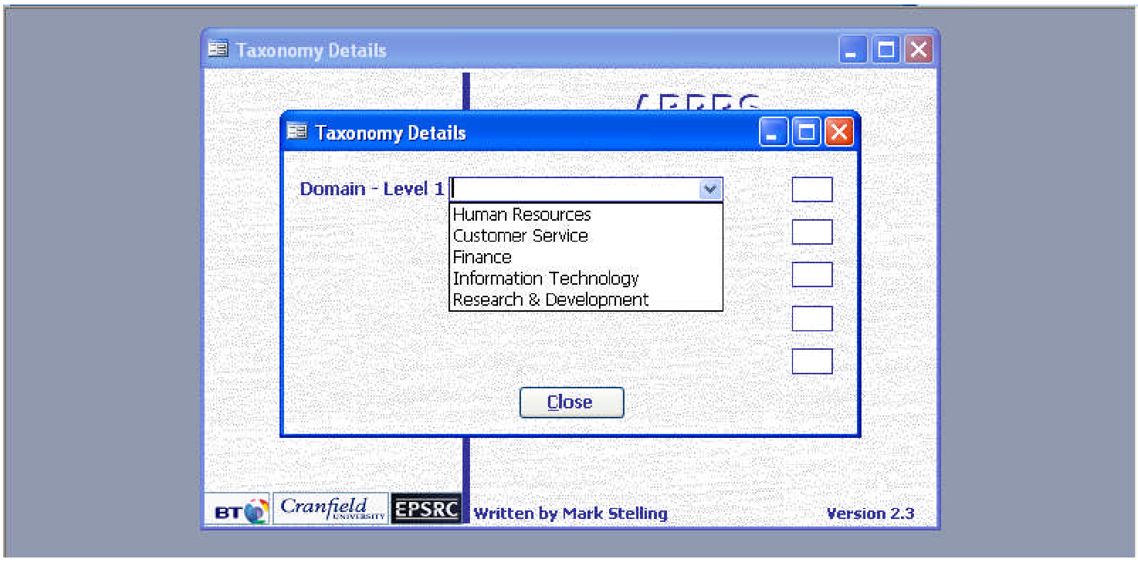


Figure 67: ABPRS - Taxonomy Details

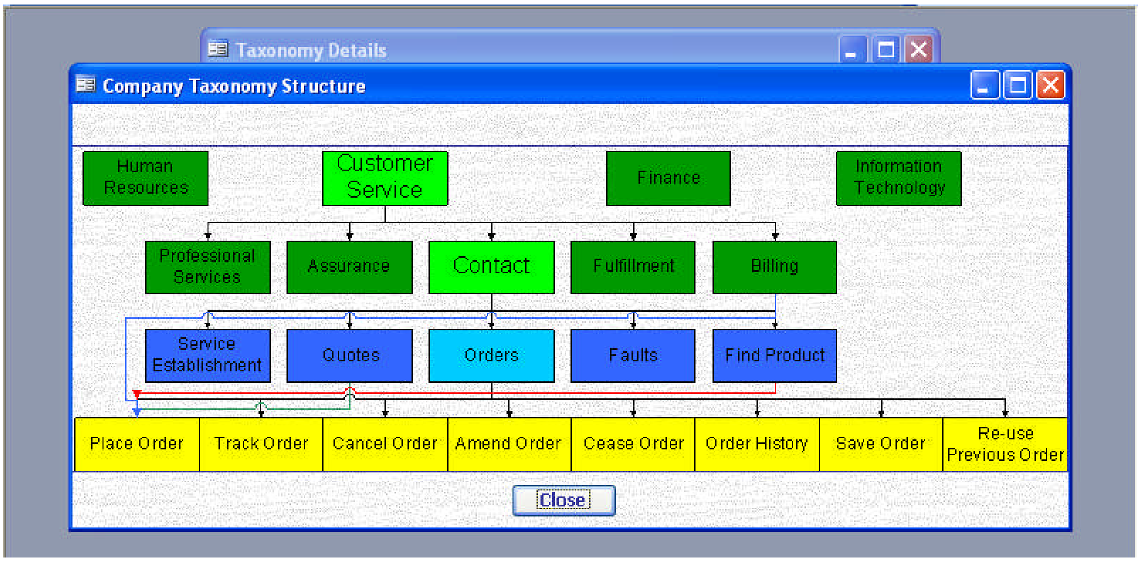


Figure 68: ABPRS - Company Taxonomy Diagram

8.2.4. Process Details

This sub-menu is shown below in Figure 69:

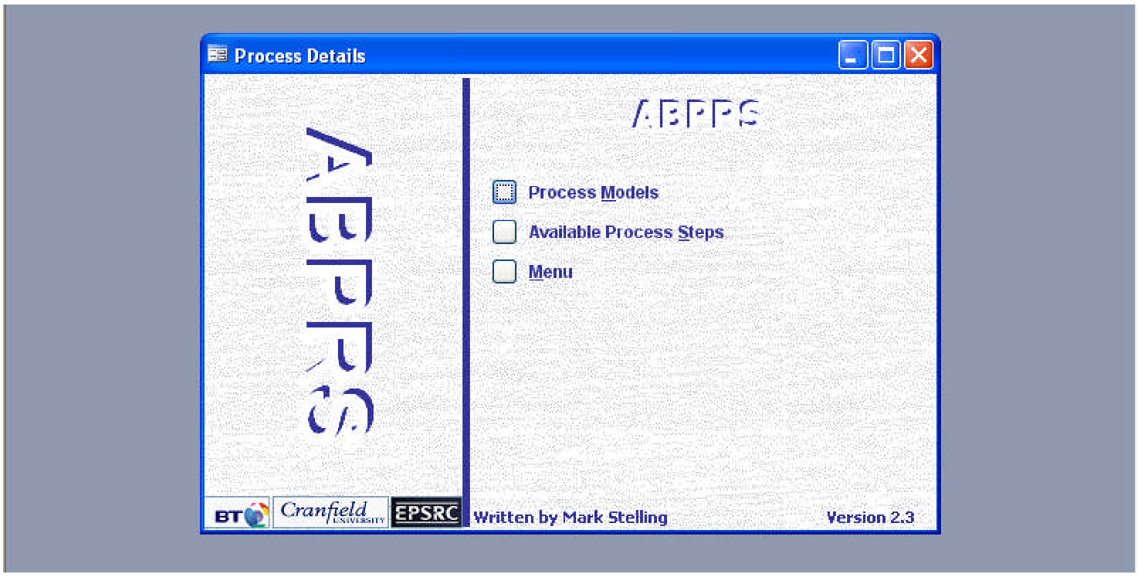
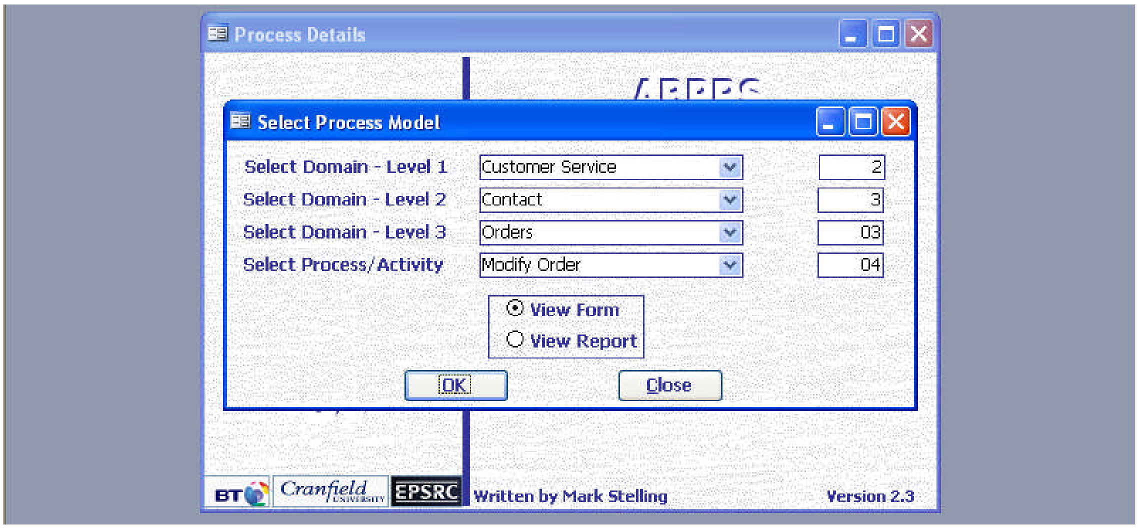


Figure 69: ABPRS - Process Details Menu

This allows the user to inspect all available process data and also to view all existing process models. Process Models are selected using the following form, Figure 70:

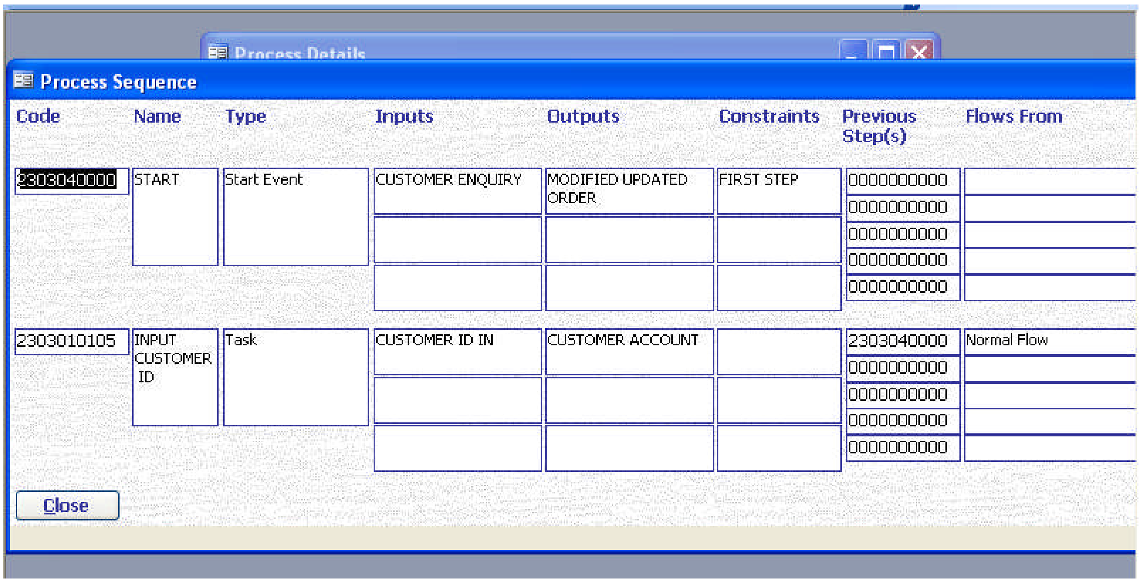


Select Domain - Level 1	Customer Service	2
Select Domain - Level 2	Contact	3
Select Domain - Level 3	Orders	03
Select Process/Activity	Modify Order	04

● View Form
○ View Report

Figure 70: ABPRS - Process Model Selection

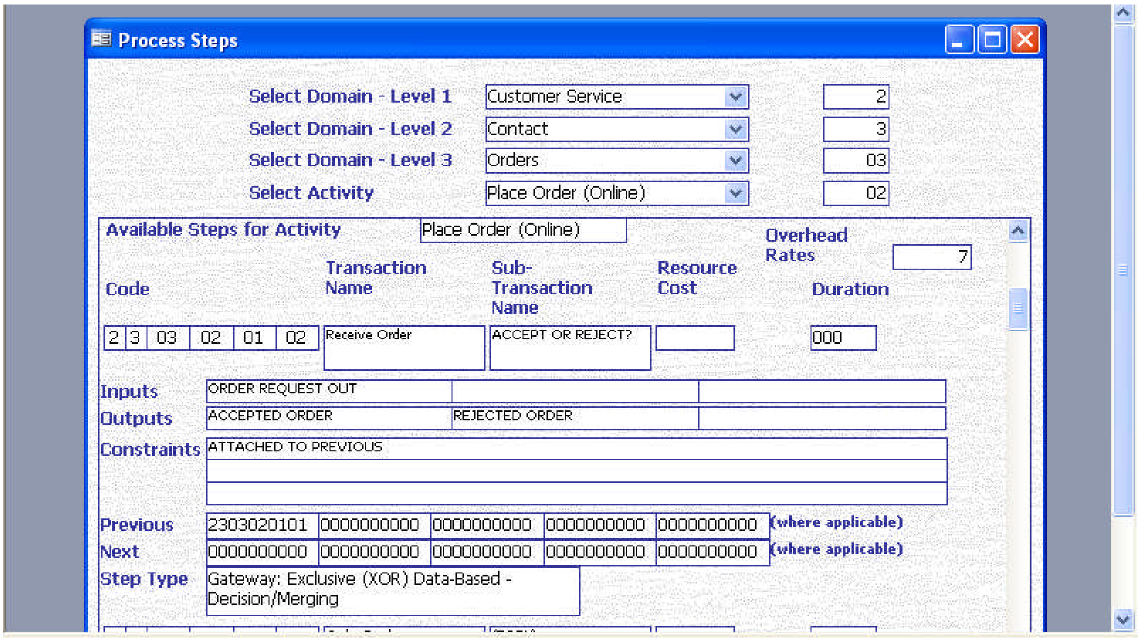
On selection of a process model, the user is then presented with a form or report detailing all elements of the model, with the coded process representation decoded. A snapshot of this is shown in Figure 71.



Code	Name	Type	Inputs	Outputs	Constraints	Previous Step(s)	Flows From
2303040000	START	Start Event	CUSTOMER ENQUIRY	MODIFIED UPDATED ORDER	FIRST STEP	0000000000 0000000000 0000000000 0000000000 0000000000	
2303010105	INPUT CUSTOMER ID	Task	CUSTOMER ID IN	CUSTOMER ACCOUNT		2303040000 0000000000 0000000000 0000000000 0000000000	Normal Flow

Figure 71: ABPRS - Process Model Form/Report

The other option on the sub-menu, Available Process Steps, allows the user to navigate down the taxonomy structure to the activity level. This then gives details of all process steps that are present in the levels below. The form allows the user to scroll through all of these and edit details if required. An example is shown in Figure 72.



Select Domain - Level 1: Customer Service 2
Select Domain - Level 2: Contact 3
Select Domain - Level 3: Orders 03
Select Activity: Place Order (Online) 02

Available Steps for Activity: Place Order (Online)
Overhead Rates: 7

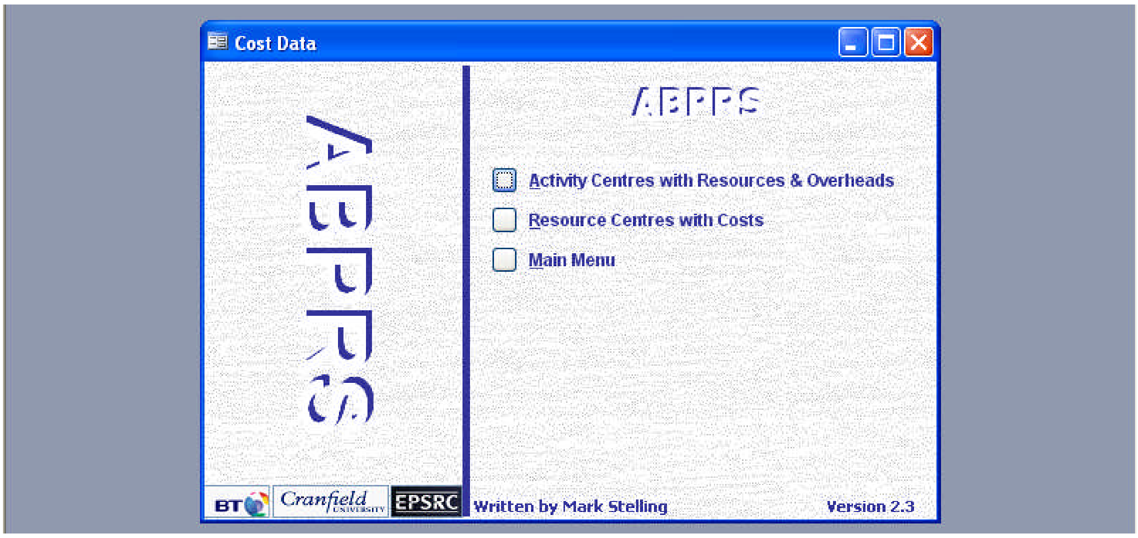
Code	Transaction Name	Sub-Transaction Name	Resource Cost	Duration
2 3 03 02 01 02	Receive Order	ACCEPT OR REJECT?		000

Inputs: ORDER REQUEST OUT
Outputs: ACCEPTED ORDER | REJECTED ORDER
Constraints: ATTACHED TO PREVIOUS
Previous: 2303020101 | 0000000000 | 0000000000 | 0000000000 | 0000000000 (where applicable)
Next: 0000000000 | 0000000000 | 0000000000 | 0000000000 | 0000000000 (where applicable)
Step Type: Gateway: Exclusive (XOR) Data-Based - Decision/Merging

Figure 72: ABPRS - Available Process Step Form

8.2.5. Cost Data

This sub-menu is shown below in Figure 73:

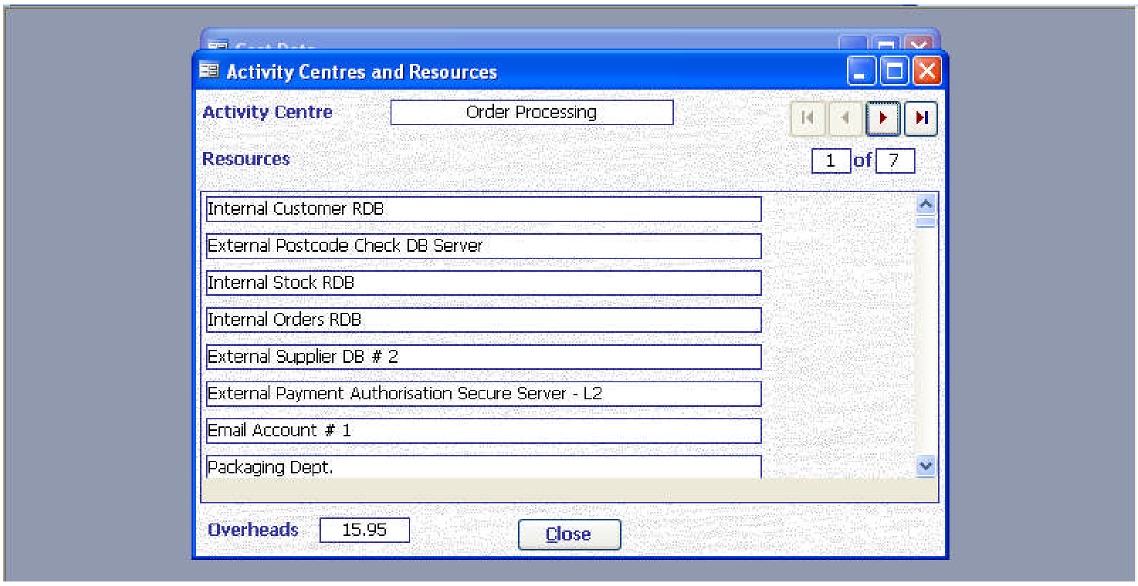


□ Activity Centres with Resources & Overheads
□ Resource Centres with Costs
□ Main Menu

Written by Mark Stelling
Version 2.3

Figure 73: ABPRS - Cost Data Menu

This sub-menu allows the user to examine all cost data. Activity centres and their resources and overhead rates can be displayed and edited if required. Figure 74 shows activity centre, Order Processing, with its overhead rates and some of its resources:



Activity Centre: Order Processing

1 of 7

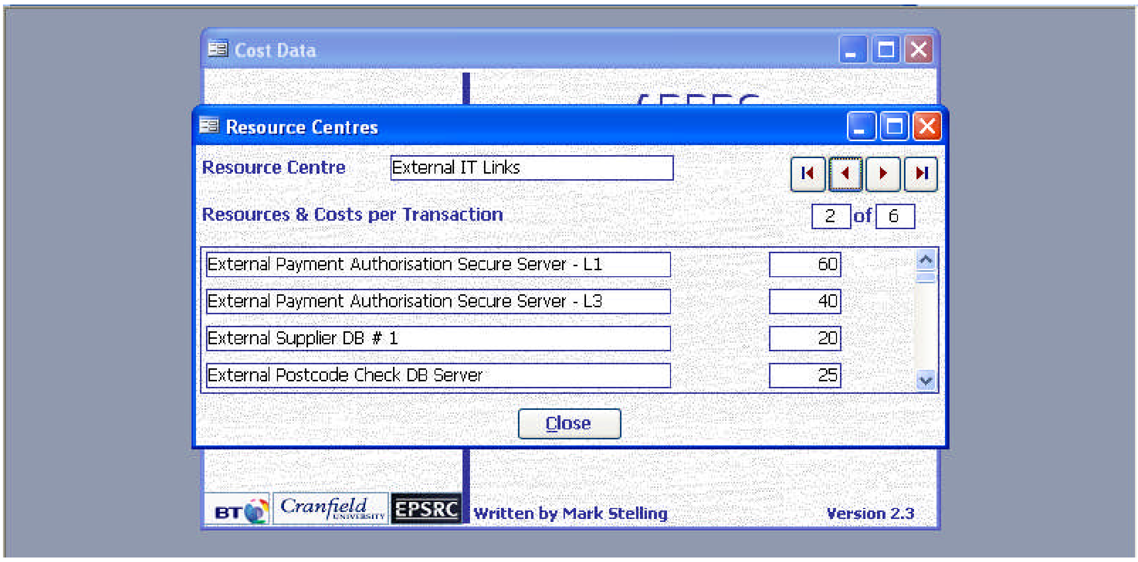
Resources

Internal Customer RDB
External Postcode Check DB Server
Internal Stock RDB
Internal Orders RDB
External Supplier DB # 2
External Payment Authorisation Secure Server - L2
Email Account # 1
Packaging Dept.

Overheads: 15.95

Figure 74: ABPRS - Activity Centre Costs Form

Resource centres with resources and costs can also be viewed and edited if required. An example, External IT Links, is shown below in Figure 75:



Resource Centre: External IT Links

2 of 6

Resources & Costs per Transaction

External Payment Authorisation Secure Server - L1	60
External Payment Authorisation Secure Server - L3	40
External Supplier DB # 1	20
External Postcode Check DB Server	25

Figure 75: ABPRS - Resource Centre Costs Form

8.3. Summary

This chapter has presented the integration of the automated BPR approach: the software platform developed to achieve this has been overviewed, including the users, user interface, inputs, outputs and database design, and the objectives and constraints and how they are all linked and used to allocate process fitness has been explained. Finally, a tour through the system has been provided, with screen prints of all menus and forms. The next chapter goes on to present a number of test scenarios for purposes of validation of the integrated technique. This is followed by validation via expert opinion.

THIS PAGE IS INTENTIONALLY LEFT BLANK

9. Results and Validation

This chapter presents the validation of the research project. A number of real-life and generic test scenarios are demonstrated in order to show the value and accuracy of the integrated reengineering approach. Expert opinion of the research is then presented, based on feedback from presentation and questionnaire concerning the outcomes of the research.

9.1. Test Scenarios

A number of test scenarios of varying complexity, involving both generic and real-life processes, are now presented. All processes are taken from the sample process library presented in Chapter 4, which are stored in the relational database described in Chapter 8. It is intended that the examples should demonstrate the intended functionality of the automated reengineering approach via the prototype software platform developed for this purpose. As detailed in Chapter 5, the algorithm performs the following four procedures, based on some of the guidelines of BPR, in an attempt to reengineer a process:

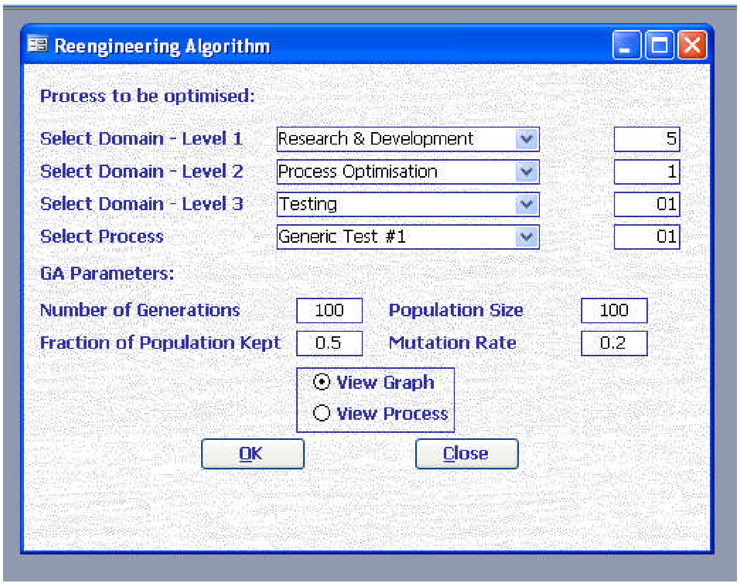
- Search for and removal of duplicates (reengineering)
- Replacement of sections of process with one step (reengineering)
- Replacement of sections of process with sub process (reengineering)
- One for one replacements of process steps with steps of differing cost and duration (optimisation)

It generates at least nine processes during the first three of the above steps, as follows:

1. Duplicates/redundant steps removed from existing process
2. Section of process replaced by a single step (forward search)
3. Section of process replaced by a single step (backward search)
4. Section of process 2 (above) replaced by sub-process (forward search)
5. Section of process 2 (above) replaced by sub-process (backward search)

6. Section of process 3 (above) replaced by sub-process (forward search)
7. Section of process 3 (above) replaced by sub-process (backward search)
8. Section of process output from 1 (above) replaced by sub-process (forward search)
9. Section of process output from 1 (above) replaced by sub-process (backward search)

Each of the above reengineered processes may then be optimised by the fourth procedure, one for one replacement of process steps via the GA based module of the technique. This only takes place on different processes: it is unlikely that nine or more different processes will be created as the nine searches/replacements are designed to cater for every possible eventuality; it is much more likely that different processes will be created by 1, 2, 4 and/or 8, but at least one will always be available (i.e. the existing process). Also, this procedure only takes place if alternative steps have been identified for the process – if no replacements are available, it is unnecessary. Before running the automated reengineering algorithm, it is necessary to supply some settings for the GA based optimisation module. The following screen print, Figure 76, illustrates these.



Process to be optimised:

Select Domain - Level 1	Research & Development	5
Select Domain - Level 2	Process Optimisation	1
Select Domain - Level 3	Testing	01
Select Process	Generic Test #1	01

GA Parameters:

Number of Generations	100	Population Size	100
Fraction of Population Kept	0.5	Mutation Rate	0.2

● View Graph
○ View Process

Figure 76: Software Platform – GA Settings

The settings, number of generations, population size, fraction of population kept and mutation rate, have been fine tuned during the course of numerous tests. The number of generations, population size, and particularly mutation rate, are set at a higher level than some of the real-life test scenarios require. However, the generic scenario and its sub scenarios, with large numbers of available one for one replacement steps in conjunction with longer processes, require these settings to be high – this is demonstrated later in the chapter.

As previously described in Chapter 5, the number of crossovers and mutations that take place per generation are calculated using the GA parameters as follows:

Number of Crossovers =
*(Population Size – ((Fraction of Population Kept * Population Size) Rounded)) / 2 Rounded*

Number of Mutations =
*((Population Size -1) * Mutation Rate) Rounded*

Therefore:

Number of Crossovers =
*(100 – ((0.5 * 100) Rounded)) / 2 Rounded = 25*

Number of Mutations =
*((100 -1) * 0.2) Rounded = 20*

Uniform Crossover (see Chapter 5) takes place between process chromosomes which are selected using Roulette Wheel Selection (see Chapter 5) – the higher the allocated fitness of a process chromosome, the higher its chance of selection for crossover. These are the ‘parents’ – 25 ‘mothers’ and 25 ‘fathers’ are selected in this case. The resultant ‘offspring’ or ‘children’ are placed in the section of the population based on the ‘fraction of population kept’ value supplied (i.e. 0.5 of 100 are kept per generation, therefore the last 50 are used to place the offspring in).

Mutation takes place 20 times per generation in this case. Chromosomes are randomly selected from the entire population, and a gene from each of these chromosomes is also randomly selected. For each mutation, a gene (or process step) is then selected from potential replacement steps for existing process steps – these may or may not be present in the current population. Only steps that have been identified as potential replacements for the selected gene can be used in order to maintain the feasibility of the process chromosome (i.e. the locus cannot change).

The following test scenarios demonstrate the results of each of the four procedures, individually and in combination. ABPRS was executed ten times for each of the test scenarios; the output from the GA based module was analysed for each scenario and the average number of generations required to converge on the optimum process was calculated; the results from the executions nearest to the average number of generations were selected for the results presented in this section. The first four scenarios are real-life, order related and credit check processes, based on those provided by the industrial sponsor (see Chapter 4 for process library details). The various removals/replacements are not necessarily realistic and are used only to demonstrate the functionality of the technique.

9.1.1. Test Scenario 1: Modify Order Process

The process Modify Order is shown below, Figure 77, in its existing state with its resource costs and probabilities underneath. Overhead costs are 0.13 per unit of duration.

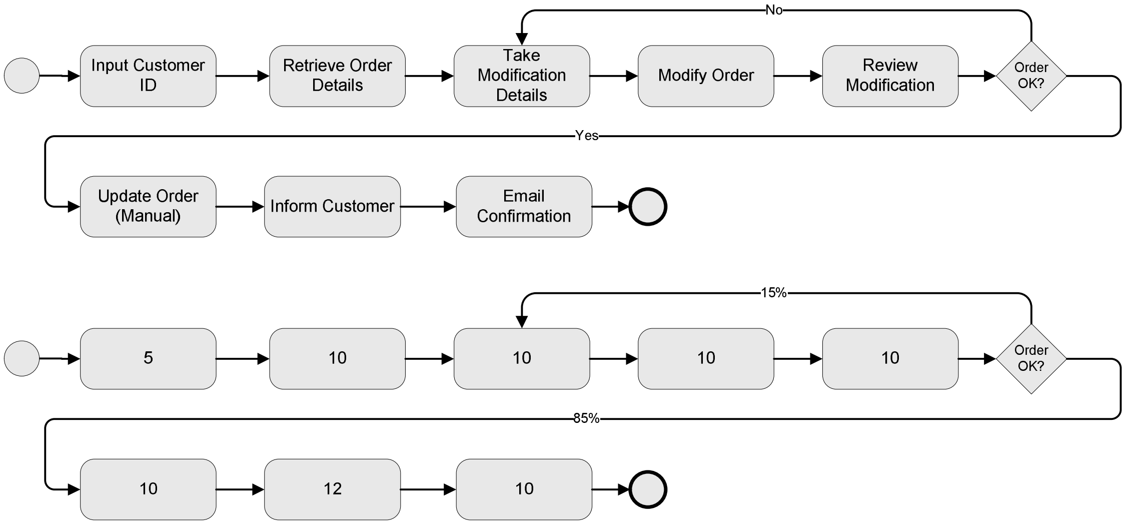


Figure 77: Test Scenario 1 – Existing Modify Order Process

This scenario involves the removal of a duplicate task step, the replacement of two task steps with one, the replacement of one task step with one of lower cost out of 2 possible alternatives, and the task step previously removed as a duplicate being used as a replacement for the other duplicate due to lower cost.

The process is shown again below, Figure 78, with an indication of its optimum state, depicted by dashed lines: ‘Modify Order Details’ is a lower resource cost/duration alternative for two steps, ‘Take Modification Details’ and ‘Modify Order’; ‘Update Order (Auto)’ is a lower resource cost/same duration alternative for the step ‘Modify Order (Manual)’; and ‘Email Confirmation’ has one matching input and output code with ‘Inform Customer’ and is therefore classed as a duplicate step, although it has a lower resource cost than ‘Inform Customer’.

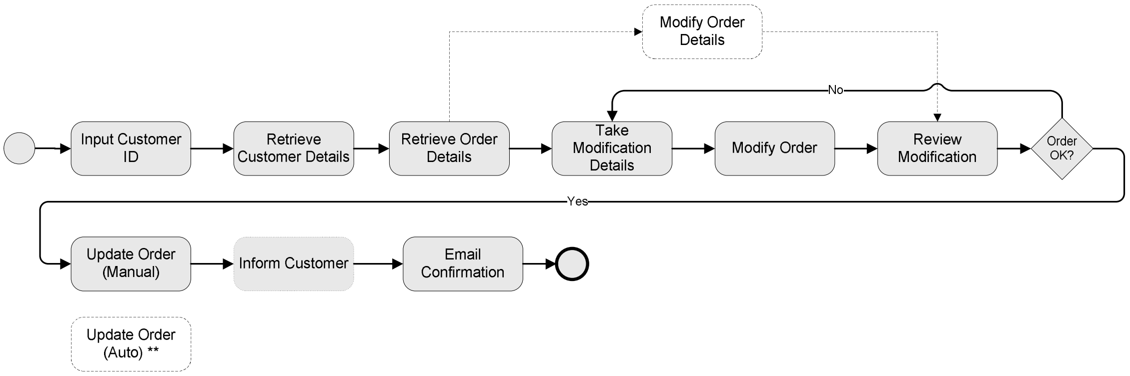


Figure 78: Test Scenario 1 –Modify Order Process Before

The stages of reengineering for this process are as follows:

- Existing Process

Cost Estimate	95.09
Duration Estimate	104.5
Customer Satisfaction	0
Fitness	0

- Search for and removal of duplicates: Remove *Email Confirmation (Cost 10 Duration 20)*
  = **Reengineered Process 1**

Cost Estimate	82.49
Duration Estimate	84.5
Customer Satisfaction	1.849
Fitness	14.449

- Replacement of sections of process with one step: Replace *Take Modification Details (Cost 10 Duration 10)* and *Modify Order (Cost 10 Duration 10)* with *Modify Order Details (Cost 10 Duration 10)*
  = **Reengineered Process 2**

Cost Estimate	69.49
Duration Estimate	73
Customer Satisfaction	2.912
Fitness	28.507

- Replacement of sections of process with sub process: None

- **Reengineered Process 1** input to optimisation stage:
  Replace *Update Order (Manual) (Cost 10 Duration 10) w*ith *Update Order (Auto) (Cost 8 Duration 10)*
  Replace *Inform Customer (Cost 12 Duration 20)* with *Email Confirmation (Cost 10 Duration 20)*

Cost Estimate	78.49
Duration Estimate	84.5
Customer Satisfaction	1.849
Fitness	18.449

- **Reengineered Process 2** input to optimisation stage:
  Replace *Update Order (Manual) (Cost 10 Duration 10) w*ith *Update Order (Auto) (Cost 8 Duration 10)*
  Replace *Inform Customer (Cost 12 Duration 20)* with *Email Confirmation (Cost 10 Duration 20)*

Cost Estimate	65.49
Duration Estimate	73
Customer Satisfaction	2.912
Fitness	32.507

The following reengineered process, Figure 79, is generated as a result:

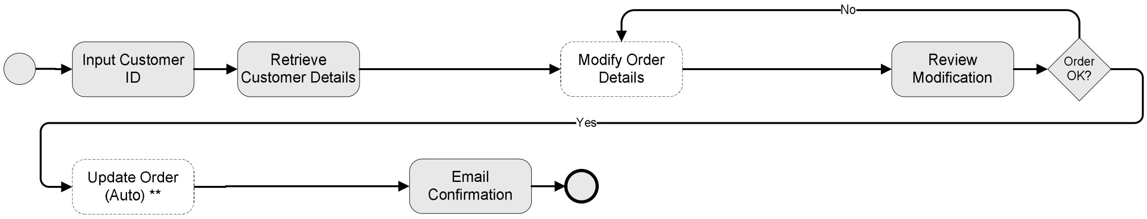


Figure 79: Test Scenario 1 –Modify Order Process After

The following chart, Figure 80, is generated in MS Excel detailing changes in estimated process cost during the course of the GA based module. It shows the cost of the original process and reengineered processes input to the optimisation stage.

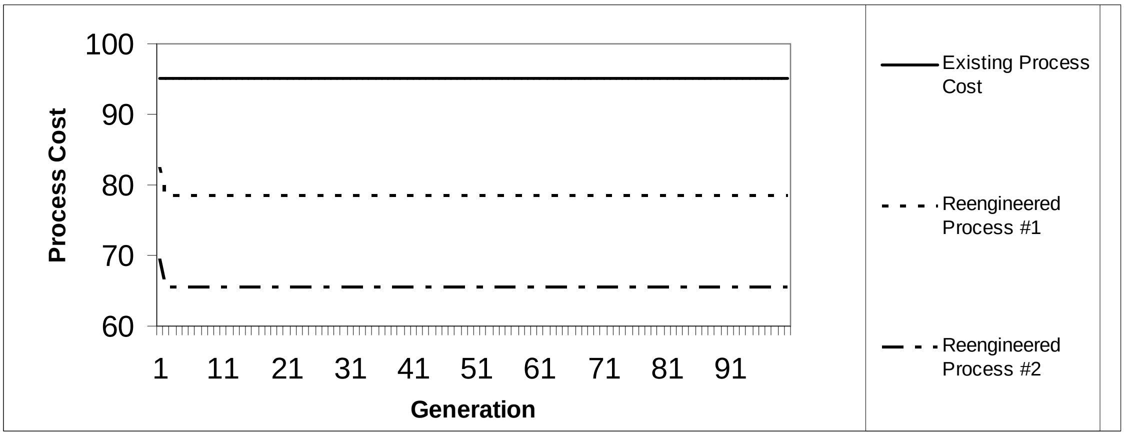


Figure 80: Test Scenario 1 – Changing Cost per Process Input to Optimisation Stage

9.1.2. Test Scenario 2: Credit Application Process

The process Credit Application is shown below, Figure 81, in its existing state with its resource costs and probabilities underneath. Overhead costs are 0.119 per unit of duration.

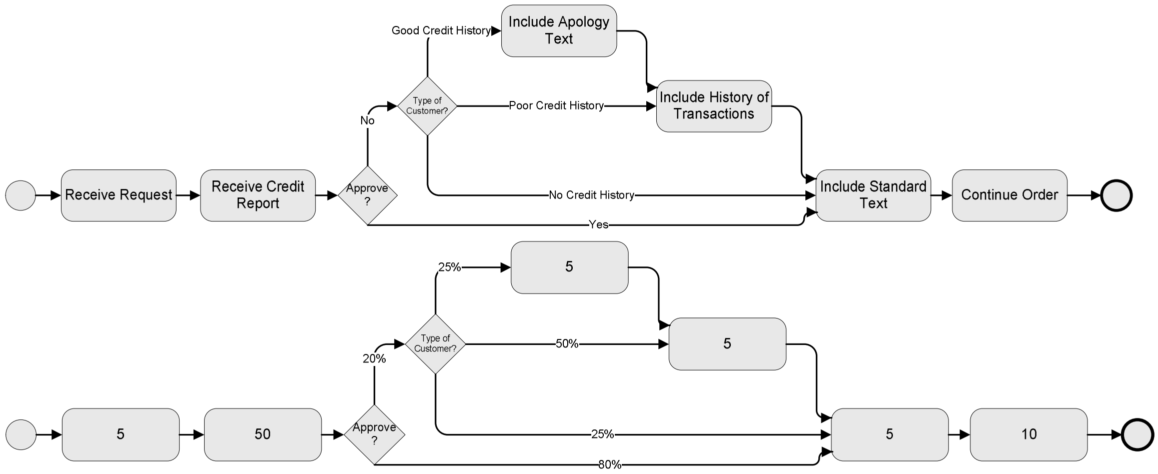


Figure 81: Test Scenario 2 – Existing Credit Application Process

This scenario involves the replacement of a large section of the process with a sub-process of higher cost, followed by the replacement of one task step with one of lower cost out of 2 possible alternatives – if this replacement didn’t take place, the reengineered process would be rejected as it would have a higher cost than the existing process. The reengineering also removes a gateway step which has a positive impact on customer satisfaction.

The process is shown again below in Figure 82 with an indication of its optimum state, depicted by dashed lines: Section ‘A’ of the process can be replaced by lower cost/duration sub-process ‘B’; and ‘External Credit Check Provider 2’ is a lower resource cost/same duration alternative for the step ‘External Credit Check Provider 3’ of the sub-process.

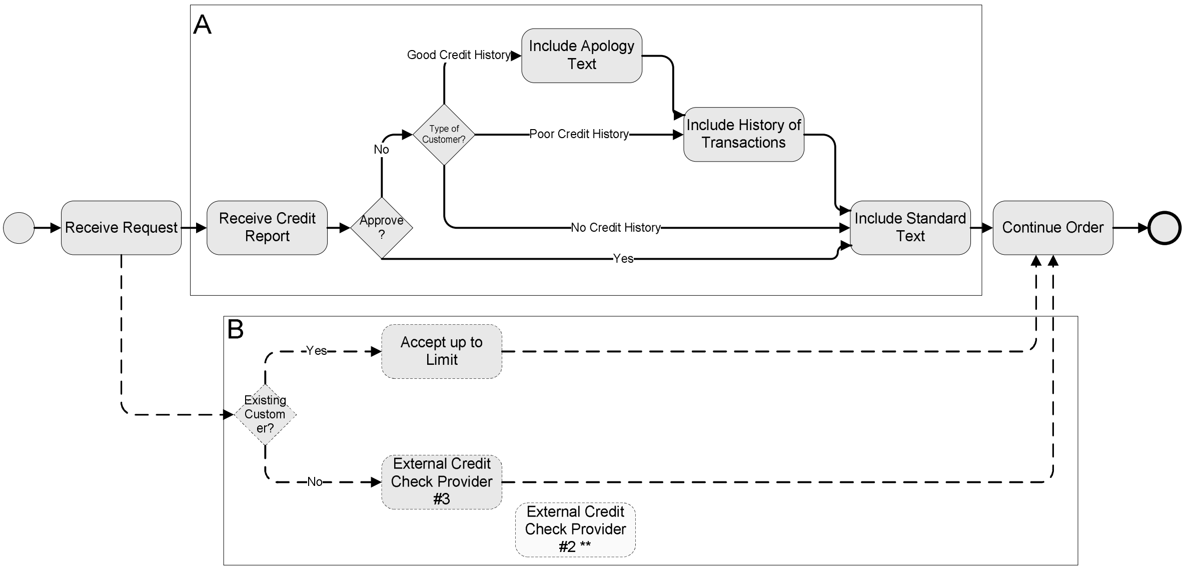


Figure 82: Test Scenario 2 –Credit Application Process Before

The stages of reengineering for this process are as follows:

- Existing Process

Cost Estimate	82.9
Duration Estimate	100
Customer Satisfaction	0
Fitness	0

- Search for and removal of duplicates: None – existing process therefore = **Reengineered Process 1**

Cost Estimate	82.9
Duration Estimate	100
Customer Satisfaction	0
Fitness	0

Replacement of sections of process with one step: None

- Replacement of sections of process with sub process: Replace Section ‘A’ of **Reengineered Process 1** with sub-process ‘B’

= **Reengineered Process 8**

Cost Estimate	105.23
Duration Estimate	86
Customer Satisfaction	1.664
Fitness	-20.669

- **Reengineered Process 1** input to optimisation stage: rejected as no available alternatives

- **Reengineered Process 8** input to optimisation stage:
  Replace *External Credit Check Provider 3 (Cost 100 Duration 100) w*ith *External Credit Check Provider 2 (Cost 55 Duration 100)*

Cost Estimate	78.234
Duration Estimate	86
Customer Satisfaction	1.664
Fitness	28.33

The following reengineered process, Figure 83, is generated as a result:

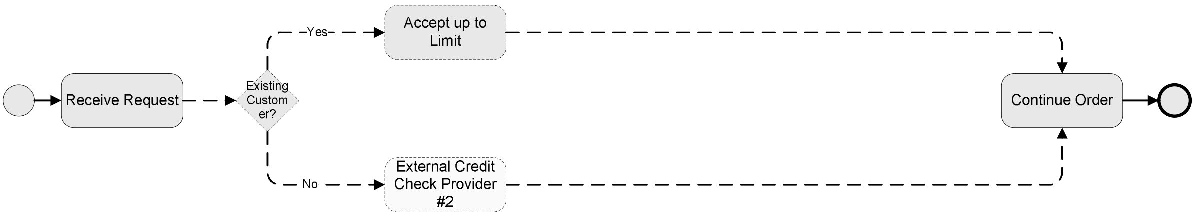


Figure 83: Test Scenario 2 –Credit Application Process After

The following chart, Figure 84, is generated - it shows the cost of the original process and the single reengineered process input to the optimisation stage.

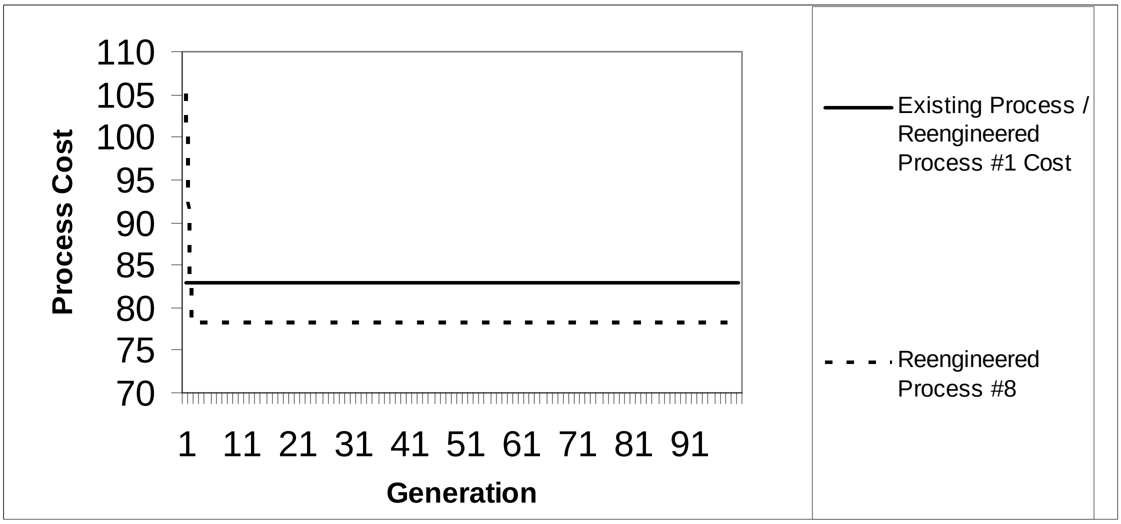


Figure 84: Test Scenario 2 – Changing Cost per Process Input to Optimisation Stage

9.1.3. Test Scenario 3: Automated/Online Order Process

The process Automated/Online Order is shown below, Figure 85, in its existing state with its resource costs and probabilities underneath. Overhead costs are 0.07 per unit of duration.

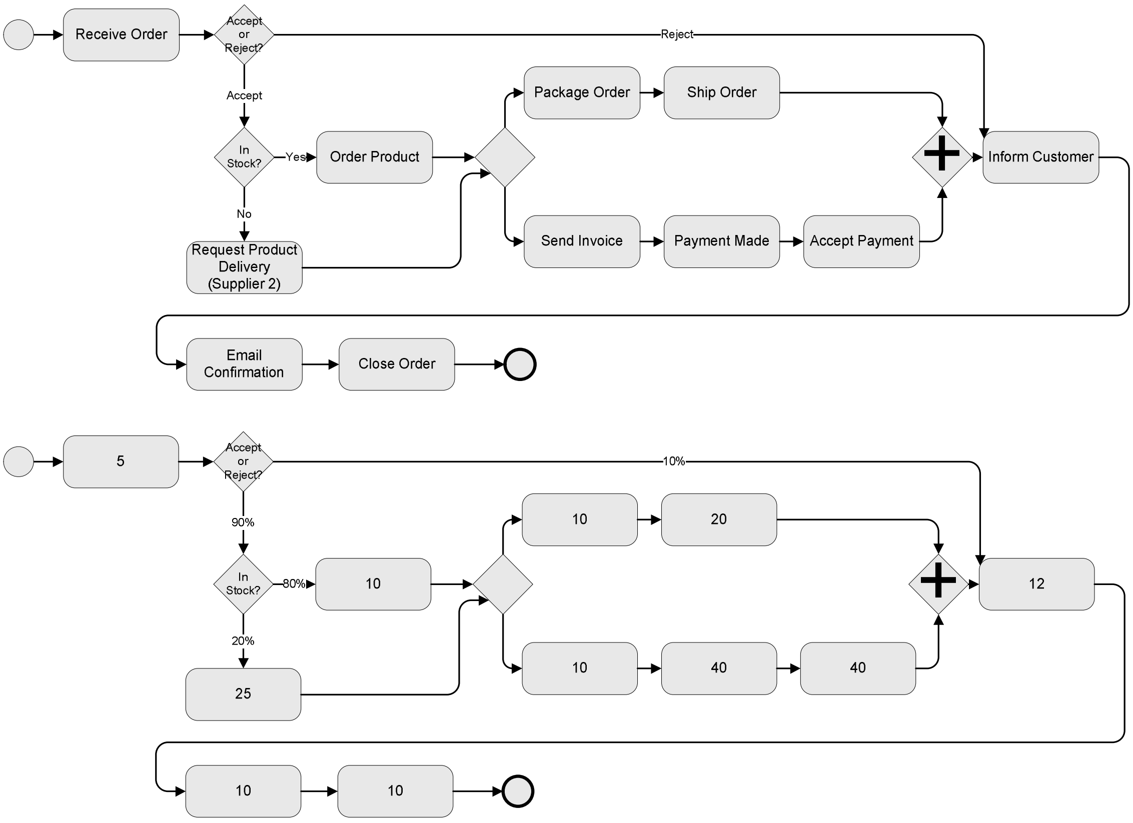


Figure 85: Test Scenario 3 – Existing Automated/Online Order Process

This scenario involves the removal of a duplicate task step, the replacement of two task steps with one, the replacement of one task step with one of lower cost out of 20 possible alternatives, and the task step previously removed as a duplicate being used as a replacement for the other duplicate due to lower cost.

The process is shown again below in Figure 86 with an indication of its optimum state, depicted by dashed lines: ‘Package and Ship EC1’ is a lower resource cost/same duration alternative for two steps, ‘Package Order’ and ‘Ship Order’, and is one of ten available replacements, ‘Package and Ship EC1’ through to ‘Package and Ship EC10’; ‘Request Product Delivery (Supplier 1)’ is a lower resource cost/same duration alternative for the step ‘Request Product Delivery (Supplier 1)’ - there are a total of 20 suppliers available in this scenario; and ‘Email Confirmation’ has one matching input and output code with ‘Inform Customer’ and is therefore classed as a duplicate step, although it has a lower resource cost than ‘Inform Customer’.

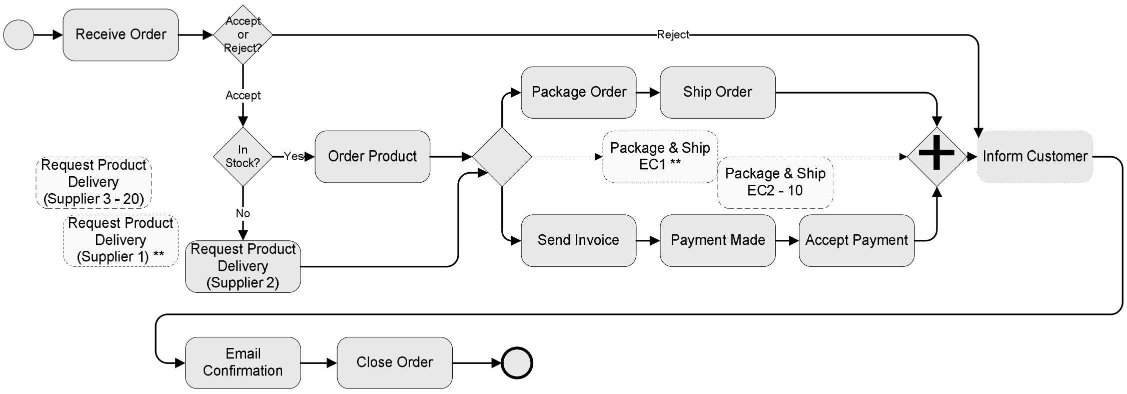


Figure 86: Test Scenario 3 –Automated/Online Order Process Before

The stages of reengineering for this process are as follows:

- Existing Process

Cost Estimate	169.78
Duration Estimate	187.1
Customer Satisfaction	0
Fitness	0

- Search for and removal of duplicates: Remove *Email Confirmation (Cost 10 Duration 20)*

  = **Reengineered Process 1**

Cost Estimate	158.38
Duration Estimate	167.1
Customer Satisfaction	0.574
Fitness	11.974

- Replacement of sections of process with one step: Replace *Package Order (Cost 10 Duration 5), Ship Order (Cost 20 Duration 5)* with one of *Package & Ship EC1 through to EC10 (Cost between 10 and 20 Duration 10)*

  = **Reengineered Process 2**

Cost Estimate	140.40 minimum
Duration Estimate	167.1
Customer Satisfaction	0.574
Fitness	29.974

- Replacement of sections of process with sub process: None

- **Reengineered Process 1** input to optimisation stage:

  Replace *Request Product Delivery (Supplier 2) (Cost 25 Duration 50)* with *Request Product Delivery (Supplier 1) (Cost 20 Duration 50)*
  Replace *Inform Customer (Cost 12 Duration 20)* with *Email Confirmation (Cost 10 Duration 20)*

Cost Estimate	155.50
Duration Estimate	167.1
Customer Satisfaction	0.574

Fitness	14.874

- **Reengineered Process 2** input to optimisation stage:

  Replace *Request Product Delivery (Supplier 2) (Cost 25 Duration 50)* with *Request Product Delivery (Supplier 1) (Cost 20 Duration 50)*

  Replace *Inform Customer (Cost 12 Duration 20)* with *Email Confirmation (Cost 10 Duration 20)*

  Replace *Package and Ship EC2 – 20 (Cost 20 Duration 10)* with *Package and Ship EC1 (Cost 10 Duration 10)* – if *Package and Ship EC1* not selected for Reengineered Process 2

Cost Estimate	137.50
Duration Estimate	167.1
Customer Satisfaction	0.574
Fitness	32.874

The following reengineered process, Figure 87, is generated as a result:

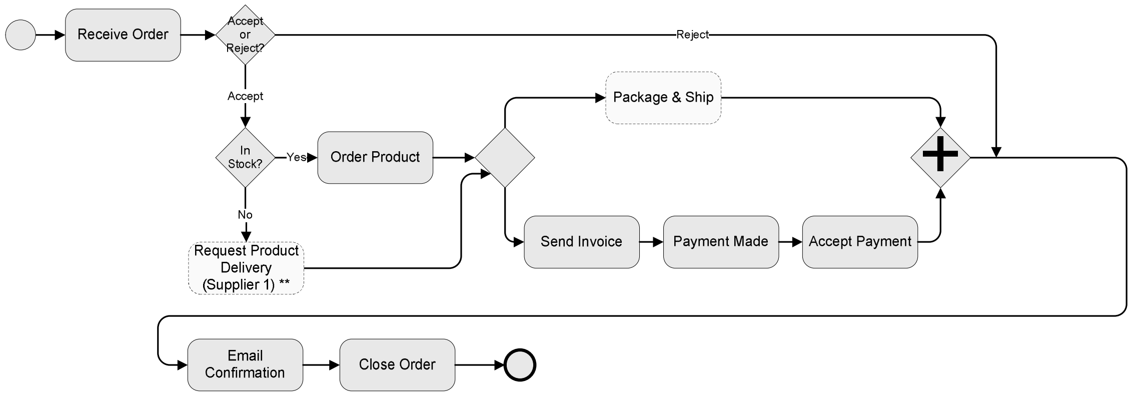


Figure 87: Test Scenario 3 –Automated/Online Order Process After

The following chart, Figure 88, is generated - it shows the cost of the original process and reengineered processes input to the optimisation stage.

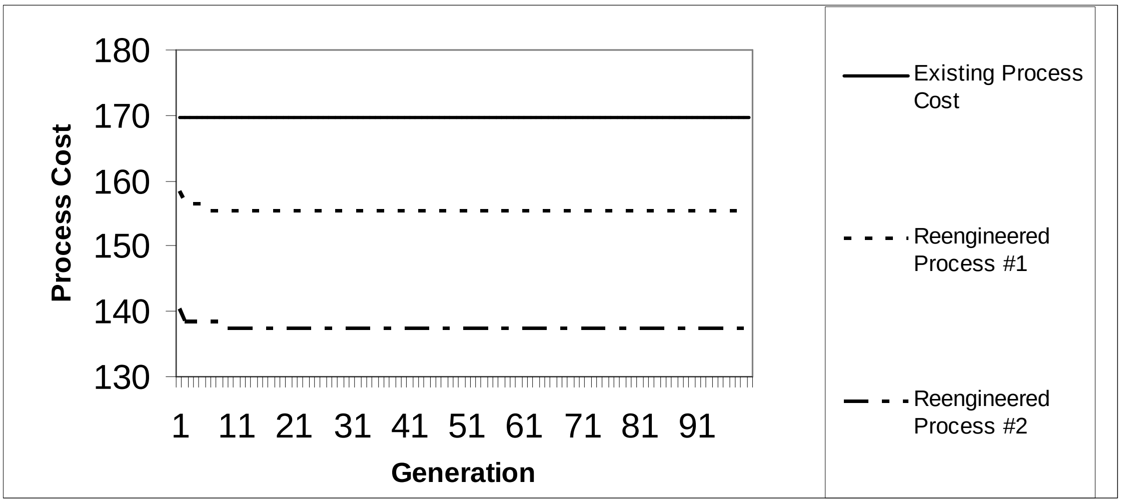
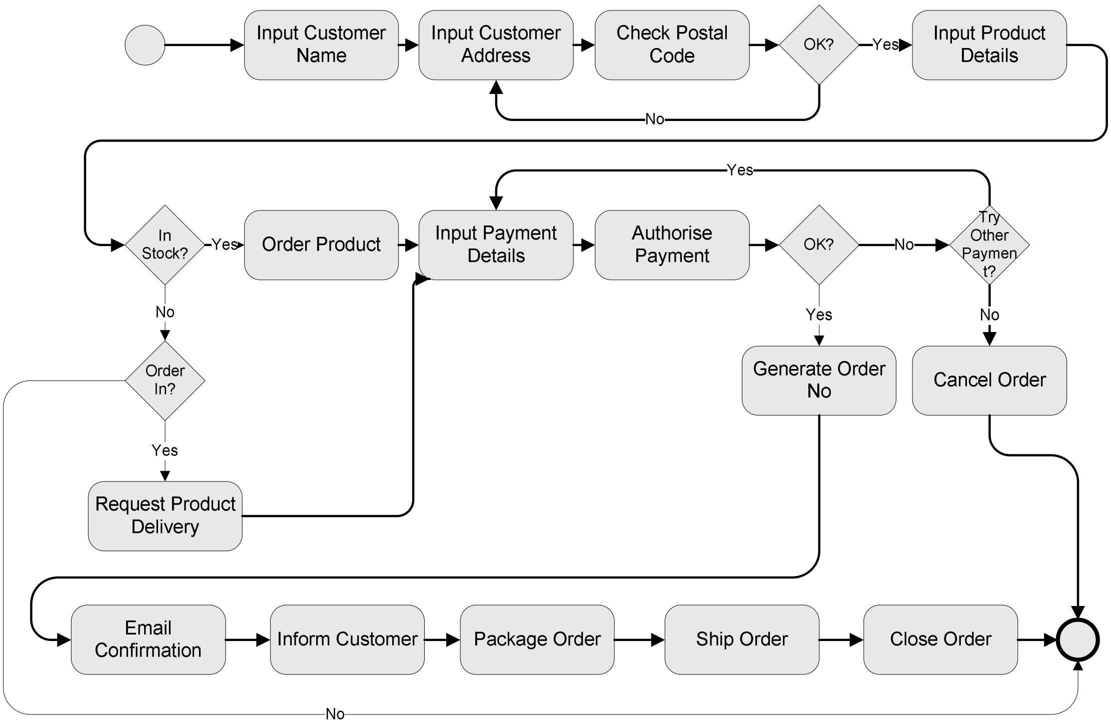


Figure 88: Test Scenario 3 – Changing Cost per Process Input to Optimisation Stage

9.1.4. Test Scenario 4: Telephone Order Process

The process Place Telephone Order is shown below, Figure 89, in its existing state with its resource costs and probabilities underneath. Overhead costs are 0.127 per unit of duration.



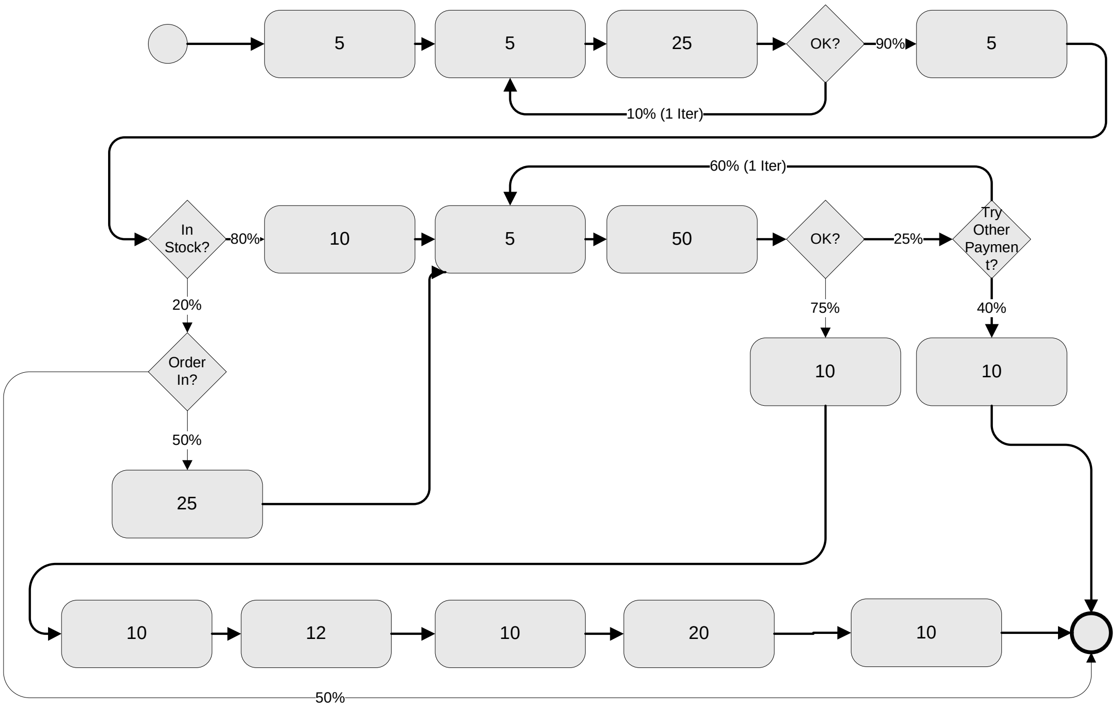


Figure 89: Test Scenario 4 – Existing Telephone Order Process

This scenario involves the removal of a duplicate task step, the replacement of four task steps with one, including the removal of a gateway step, the replacement of two task steps with one, the replacement of one task step with one of lower cost out of 20 possible alternatives, the replacement of one task step with one of lower cost out of 15 possible alternatives, and the choice of 10 replacement task steps for one step in the existing process which is already of the lowest cost. It also involves the replacement of a large part of the process with a sub process, leading to a dramatic reconfiguration of the whole process in which all gateways are removed, leaving a simple serial process.

The process is shown again below in Figure 90, with an indication of its optimum state, depicted by dashed lines: ‘Input Customer ID’ is a lower resource cost/duration alternative for four steps, ‘Input Customer Name’, ‘Input Customer Address’, ‘Check Postal Code’ and ‘OK?’; ‘Request Product Delivery (Supplier 1)’ is a lower resource cost/same duration alternative for the step ‘Request Product Delivery (Supplier 1)’ - there are a total of 20 suppliers available in this scenario; ‘Authorise Payment (L3)’ is a lower resource cost/same duration alternative for ‘Authorise Payment (L2)’ – there are 15 payment authorisers available; ‘Inform Customer’ has one matching input and output code with ‘Email Confirmation’ and is therefore classed as a duplicate step; and ‘Package and Ship EC1’ is a lower resource cost/same duration alternative for two steps, ‘Package Order’ and ‘Ship Order’, and is one of ten available replacements, ‘Package and Ship EC1’ through to ‘Package and Ship EC10’. Sub process ‘A’ can replace all steps after ‘Input Product Details’ through to ‘Close Order’.

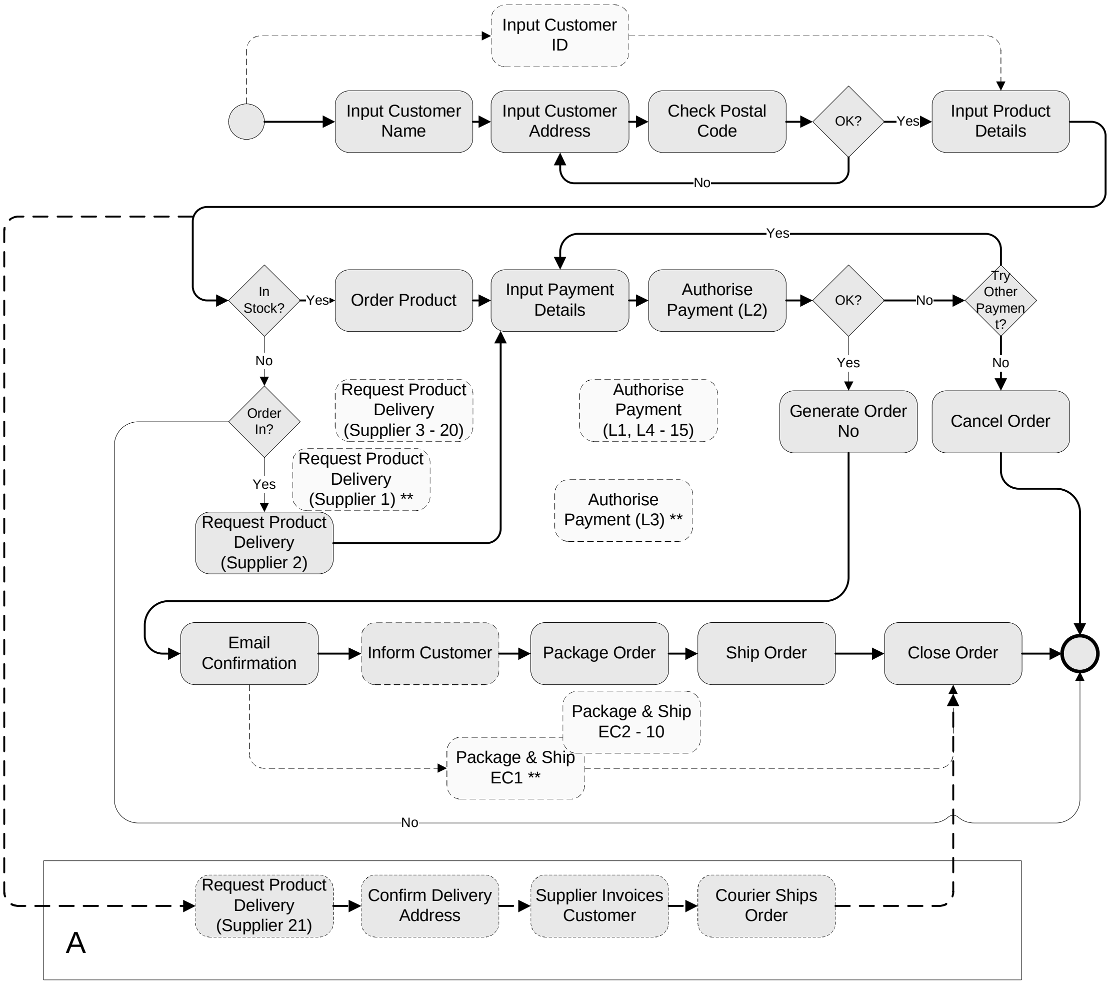


Figure 90: Test Scenario 4 –Telephone Order Process Before

The stages of reengineering for this process are as follows:

- Existing Process

Cost Estimate	195.02
Duration Estimate	276.35
Customer Satisfaction	0
Fitness	0

- Search for and removal of duplicates: Remove *Inform Customer (Cost 12 Duration 20)*

  = **Reengineered Process 1**

Cost Estimate	185.21
Duration Estimate	262.85
Customer Satisfaction	0.177
Fitness	9.992

- Replacement of sections of process with one step: Replace *Input Customer Name (Cost 5 Duration 10), Address (Cost 5 Duration 20), Check Postal Code (Cost 25 Duration 15), OK?* with *Input Customer ID (Cost 5 Duration 10)*

  Replace *Package Order (Cost 10 Duration 5), Ship Order (Cost 20 Duration 5)* with one of *Package & Ship EC1 through to EC10 (Cost between 10 and 20 Duration 10)*

  = **Reengineered Process 2**

Cost Estimate	133.82 minimum
Duration Estimate	224.35
Customer Satisfaction	0.783
Fitness	61.987

- Replacement of sections of process with sub process: Replace section of **Reengineered Process 2** with sub process 'A':

  Replace all steps after *Input Product Details* through to *Close Order* with sub process A

  = Reengineered Process 4

Cost Estimate	73.34
Duration Estimate	105
Customer Satisfaction	2.752
Fitness	124.4

- Replacement of sections of process with sub process: Replace section of **Reengineered Process 1** with sub process 'A':

  Replace all steps after *Input Product Details* through to *Close Order* with sub process A

  = Reengineered Process 8

Cost Estimate	111.22
Duration Estimate	143.5
Customer Satisfaction	2.146
Fitness	85.94

- Input **Reengineered Process 1** to optimisation stage:

  Replace *Request Product Delivery (Supplier 2) (Cost 25 Duration 50)* with *Request Product Delivery (Supplier 1) (Cost 20 Duration 50)*

  Replace *Authorise Payment (L2) (Cost 50 Duration 60)* with *Authorise Payment (L3) (Cost 40 Duration 60)*

Cost Estimate	174.36
Duration Estimate	262.85
Customer Satisfaction	0.177
Fitness	23.842

- Input **Reengineered Process 2** to optimisation stage:

  Replace *Request Product Delivery (Supplier 2) (Cost 25 Duration 50)* with *Request Product Delivery (Supplier 1) (Cost 20 Duration 50)*

  Replace *Authorise Payment (L2) (Cost 50 Duration 60)* with *Authorise Payment (L3) (Cost 40 Duration 60)*

  Replace *Package and Ship EC2 – 20 (Cost 20 Duration 10)* with *Package and Ship EC1 (Cost 10 Duration 10)* – if *Package and Ship EC1* not selected for Reengineered Process 2

Cost Estimate	122.97
Duration Estimate	224.35

Customer Satisfaction	0.783
Fitness	72.837

- **Reengineered Process 4** input to optimisation stage: rejected as no available alternatives

- **Reengineered Process 8** input to optimisation stage: rejected as no available alternatives

The following reengineered process, Figure 91, is generated as a result:

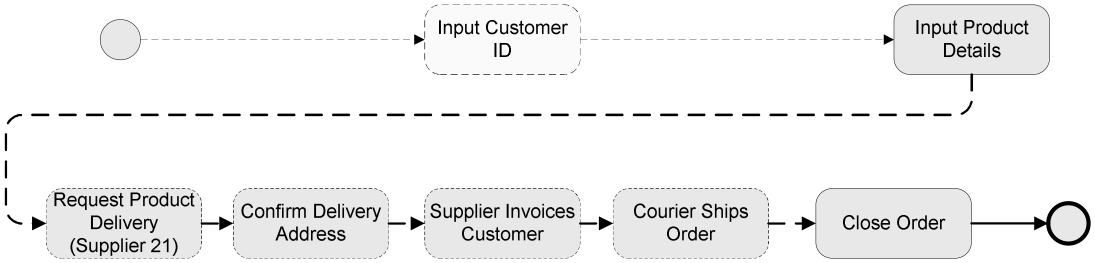


Figure 91: Test Scenario 4 –Telephone Order Process After

The following chart, Figure 92, is generated:

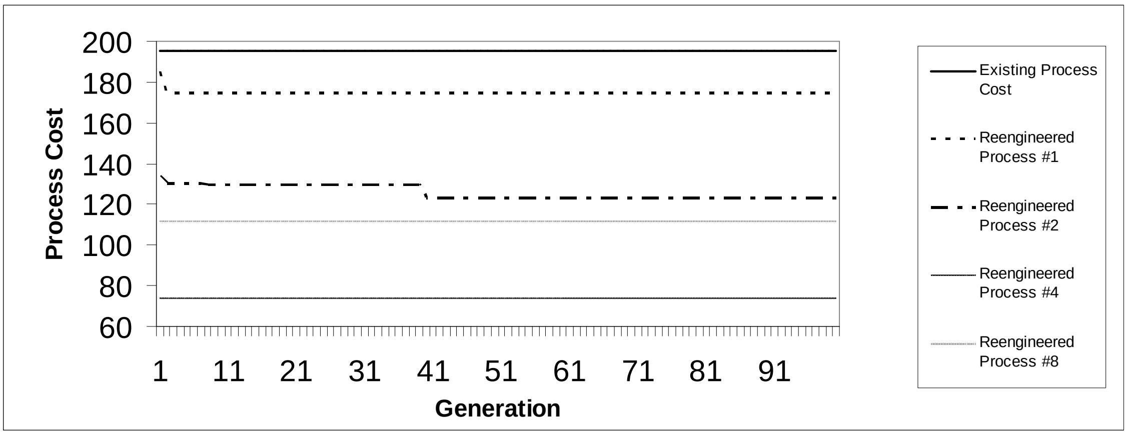


Figure 92: Test Scenario 4 – Changing Cost per Process Input to Optimisation Stage

9.1.5. Test Scenario 5: Generic Processes

In the previous test scenarios, all processes were successfully reengineered, demonstrating all of the stages involved in the technique. However, in the first two scenarios, due to the relatively small number of alternative process steps available, the GA based optimisation procedure generally converges on the optimum process in its first to third generation. This is primarily due to all or the majority of potential processes being created in the initial population. The number of available alternatives is increased in the third scenario – it takes on average six generations to converge on the optimum process for Reengineered Process No. 1 and nine generations for Reengineered Process No. 2. Due to an even larger number of available alternatives, in the fourth test scenario, the GA converges on the optimum process on average in its fifteenth generation for Reengineered Process No. 1 and its fortieth generation for Reengineered Process No 2.

The intent of the following scenarios is to further demonstrate the value of the GA based part of the automated reengineering methodology by dramatically increasing available alternative process steps – this means that relatively few potential processes are introduced into the initial populations as there are very large numbers of possibilities, and the GA therefore relies on its operators, reproduction, crossover and mutation, to eventually generate the optimum process. The desire with the generic scenarios is to take the problem complexity a step further than the fist four scenarios, even though the scenarios are not realistic. The number of existing process steps and alternatives are varied and the results are compared.

The figure below (Figure 93) shows a generic, non-specific process, derived from the 'Research & Development' domain of the company taxonomy.

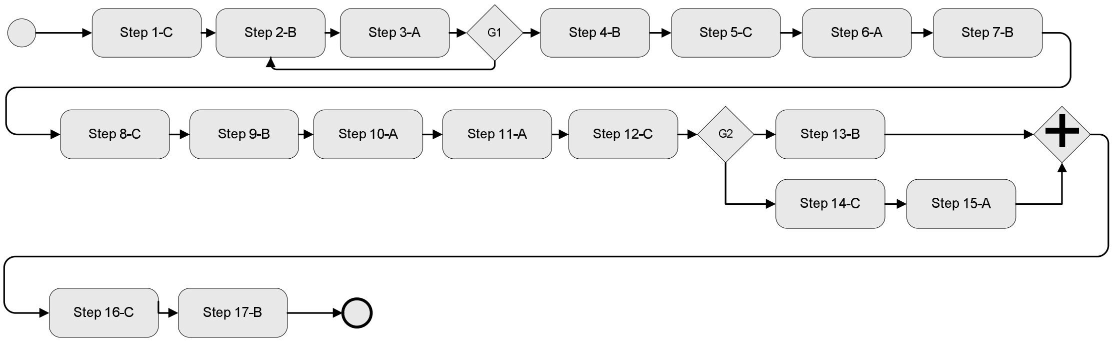


Figure 93: Test Scenario 5 – Generic Process

Six sub scenarios have been derived from the above process as follows:

1. For each of the existing 17 task steps there are two possible alternatives - an A, B and C step is available in every case, as shown in the Figure 90 below. Number of possible processes is 3^{17}.
2. For each of the 17 task steps there are four possible alternatives (an A, B, C, D and E step is available). Number of possible processes is 5^{17}.
3. There are nine possible alternatives (an A, B, C, D, E, F, G, H, I and J step is available). Number of possible processes is 10^{17}.
4. There are nineteen possible alternatives (an A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S and T step is available). Number of possible processes is 20^{17}.
5. The number of task steps is increased from 17 to 34, each step having two possible alternatives (A, B, C step). Number of possible processes is 3^{34}.
6. The number of task steps is increased to 60, each step having two possible alternatives. Number of possible processes is 3^{60}.

There is also one available step which replaces three of the existing steps, and one duplicate for each sub scenario.

The size of the initial process chromosome in the first four examples is 3828 digits containing 22 genes. The optimum process steps in all examples are the ‘A’ steps, and the duplicate (to be removed) is indicated by dotted lines. The following figure (Figure 94) shows the generic process with 3 alternatives per 17 task steps.

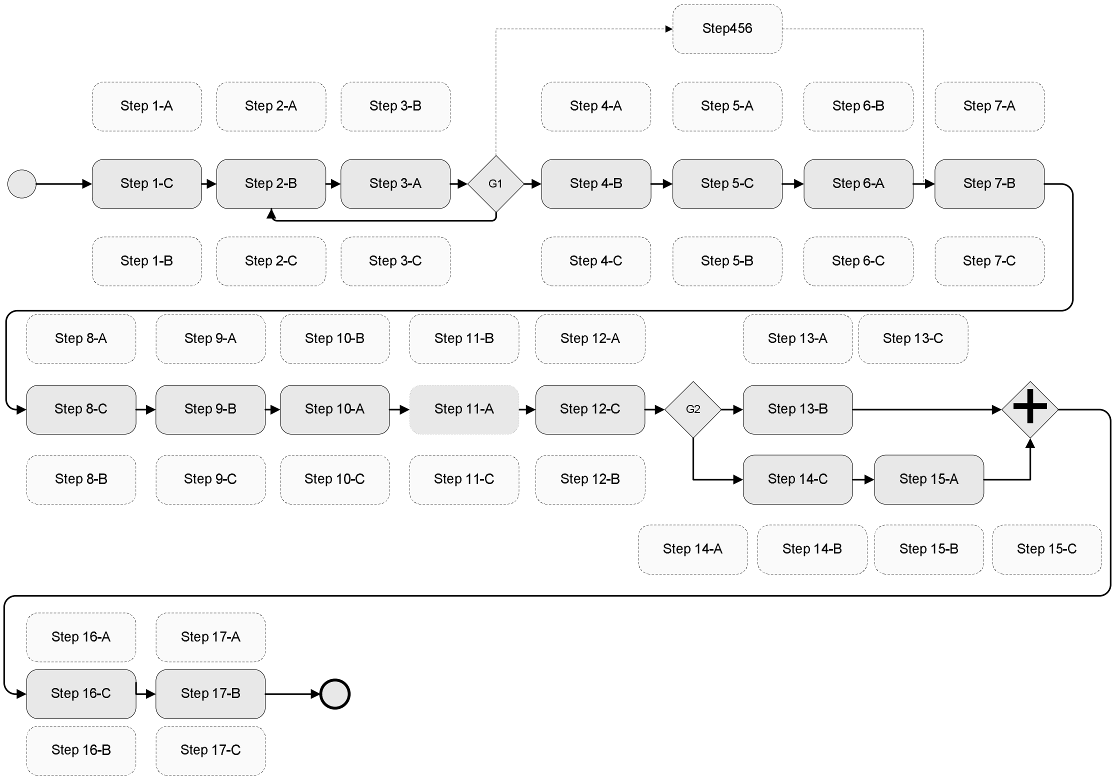


Figure 94: Test Scenario 5 –Generic Process Before

The reengineering algorithm is executed for the first four sub scenarios (3, 5, 10 and 20 available alternatives) and produces the optimum process as follows in Figure 95:

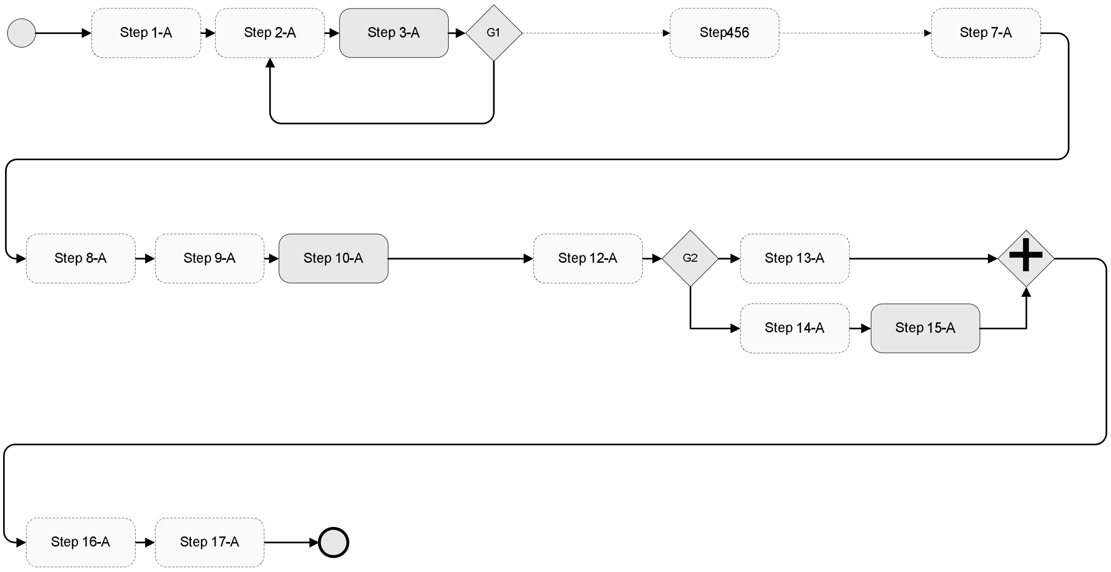


Figure 95: Test Scenario 5 –Generic Process After

As the purpose of this scenario is to demonstrate the effect of increasing the numbers of available alternatives, along with process length, on the GA based part of the methodology, only the 'best' of the reengineered processes input to the optimisation procedure is shown on the chart, Figure 96, below (i.e. after duplicate removed and section replaced with one step) – this illustrates the comparison between the varying numbers of alternatives more clearly. Also, duration and customer satisfaction are not detailed as in the real-life scenarios.

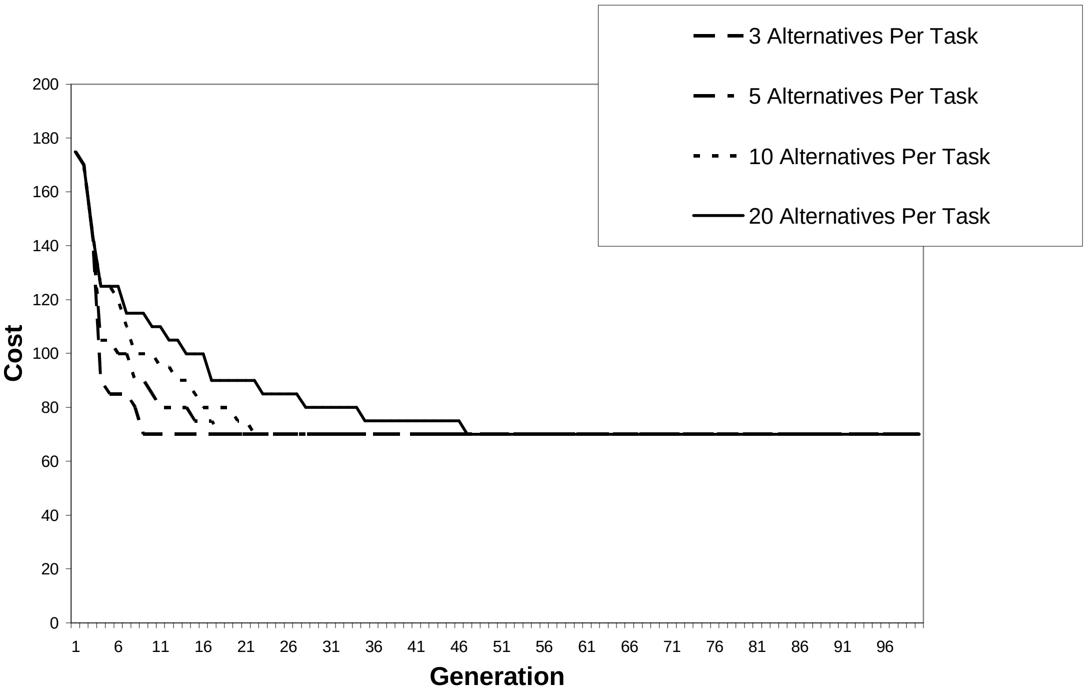


Figure 96: Test Scenario 5 –Generic Process Sub Scenario Comparison 1 to 4

It can be seen in the above chart that the number of available alternatives has a significant effect on the GA based module – the more that are available, the more generations are required to converge on the optimum process.

Then the number of process tasks is increased from 17 to 34 for the fifth sub scenario and then to 60 for the sixth – these extra steps are added sequentially to the process model with the 'A' steps again representing the optimum solution. Three alternatives are available for each of the steps. The comparison is as follows (Figure 97):

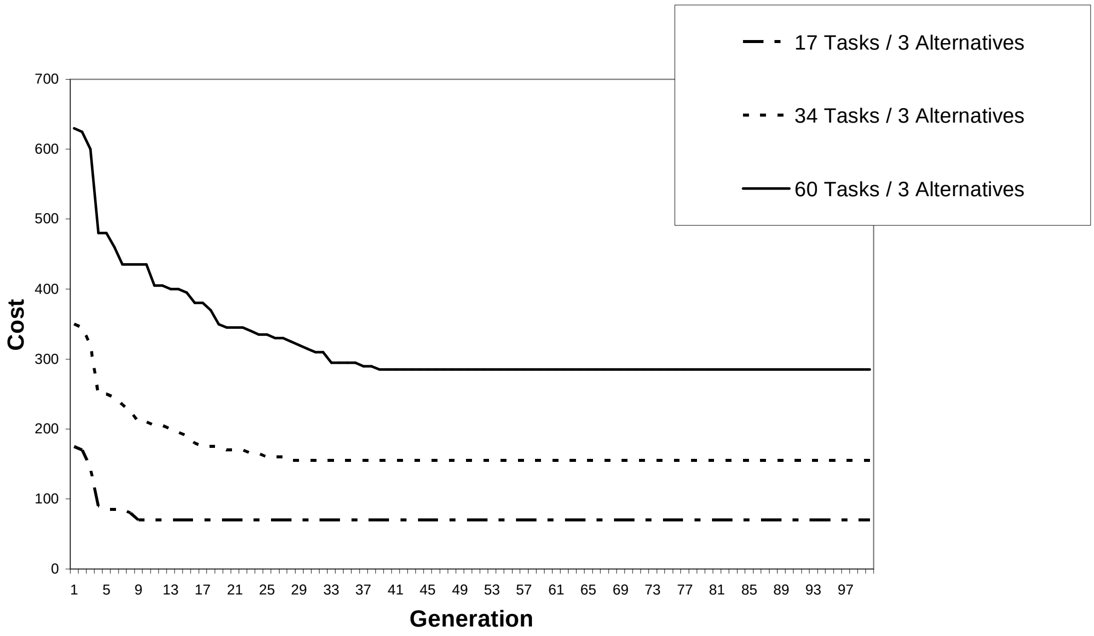


Figure 97: Test Scenario 5 –Generic Process Sub Scenario Comparison 1,5 and 6

It can be seen in the above chart that the number of tasks, as with available alternatives, has a significant effect on the GA based module – the more tasks in the process, the more generations are required to converge on the optimum.

9.2. Expert Opinion

The second phase of validation involved employing the opinion of experts. A total of six experts were presented with the research and asked to provide ratings and comments on each element of the project, and also on achievement of overall aim and objectives.

They included: two experts from BT, both senior research officers (PhD qualified) with experience in business process management and business intelligence, one who was already familiar with the project and one who had no prior knowledge of it; a senior marketing analyst (PhD qualified) specialising in customer churn management (UPS); a senior manager with experience of process monitoring (EDF/Sainsbury's Energy); a senior business process analyst (MSc qualified) specialising in monitoring of customer satisfaction (Cambridgeshire Constabulary, Citizen Experience Project); and a group financial controller (MBA qualified) specialising in ABC (ACIS). The presentation was made in sections, each dealing with the different elements of the research, and the questionnaire corresponds to the sections – after each section of the presentation was completed, the corresponding section of the questionnaire was completed. The complete questionnaire can be seen in Appendix C. A summary of the questions is listed in this section with average ratings, either from 1 to 5 (poor through to excellent) for effectiveness, accuracy and usability, or from 1 to 5 (strongly disagree, disagree, neither agree or disagree, agree, strongly agree). The facility to make comments on strengths and weaknesses, or just general explanations for the ratings chosen, was also included with each question. A respondent profile was also included. Analysis of the ratings, comments and respondents' experience enabled a critical judgement to be made of the research.

9.2.1. Interviewee Profiles

The average experience/knowledge of all interviewees in a number of relevant areas is shown in Table 18 (rating 1 to 5, or poor through to excellent):

Table 18: Validation – Average Respondent Profile

Area	Rating
Business Process Modelling/Architecture	4.67
Business Process Reengineering	4
Conventional and Evolutionary Computing	3.67
Database Design	3.5
Data Mining	4.17
Cost Estimation	4
Measuring/Monitoring Customer Satisfaction	4.5

The questions and average ratings are as follows:

9.2.2. Process Coding Mechanism

1) Taxonomy creation leading to unique identifier which allows tracing of every process step to point of origin.

2) Process coding system.

3) Translation from alternative process model (IDEF3, ARIS) to BPMN, then BPMN to codes.

Average Rating 1 – 5 (poor – excellent)			
Question No.	1	2	3
Effectiveness	4.17	4.5	4
Accuracy	4.17	4.33	4
Usability	4.17	3.67	3

Comments:

- “One of the more effective and straightforward ways in which to develop a hierarchal structure”.
- “As with any application that leads to the development of a hierarchal structure, conflicts can arise and will need to be managed”.
- “Possible usability barrier (long codes). However, User Interface design should overcome”.
- “Making graphical models amenable to processing with algorithms like GA”.
- “Need automatic translation to gain usability”.

9.2.3. Automated Reengineering

4) Stages of the technique (e.g. replicate guidelines of business process reengineering as much as possible without human expert judgement).

5) The GA module:

a) Maximises search space to enable generation of potentially every possible available process.

b) Generation of feasible processes only.

6) Example processes – realistic and/or explanatory.

7) Reengineered example output processes are superior to input.

Average Rating 1 – 5 (poor – excellent)		
Question No.	4	5b
Effectiveness	4.17	4.5
Accuracy	3.5	4.5
Usability	3.67	4.83

Average Rating 1 – 5 (strongly disagree - strongly agree)			
Question No.	5a	6	7
	5	4.17	5

Comments:

- “Comprehensive reengineering”.
- “Might involve continuous (periodic) maintenance”.
- “Does not depend on intimate knowledge of GA by user”.
- “Might need more examples”.
- “The example processes were well laid out and straightforward as examples”.
- “There is a proportion of subjectivity in business process engineering, and I am not sure that the technique would be fully able to encapsulate this”.

9.2.4. Cost Estimation

8) ABC structure (e.g. as per literature/survey).

9) Cost drivers/method of allocating costs to direct resources, charged per transaction, and then overheads charged by rate per unit of time.

10) Varying process resource costs based on probability of usage – to differentiate between processes that use the same resources, but in varying degrees.

11) Automated method of estimating process cost (e.g. virtual array, stacking of probabilities).

12) Method of allocating fitness score (e.g. Delta Cost).

Average Rating 1 – 5 (poor – excellent)					
Question No.	8	9	10	11	12
Effectiveness	5	5	4.67	5	4
Accuracy	4.67	4.67	4.83	5	4.17
Usability	3.67	4.67	4	3.67	4.67

Comments:

- ” Effective and clear to business users”.
- “Clear to business users and to financial systems”.
- “Without an adequately qualified cost estimator (or statistical analyst), some users of this information may be unable to fully utilise it due to a lack of understanding”.
- “It needs prior analysis of execution data that might not always be available”.

9.2.5. Customer Satisfaction

13) ‘Cycle Time’ and ‘Right First Time’ criteria.

14) Utilisation of all information held in process models to achieve indication of negative or positive effect.

15) This is potentially the only automated means of providing an indication without feedback from customer or customer related data.

16) The customer satisfaction formula.

17) Use of value produced by formula (e.g. slight edge in selection for process with positive; reject negative).

Average Rating 1 – 5 (poor – excellent)				
Question No.	13	14	16	17
Effectiveness	4.83	4.17	4.33	4.67
Accuracy	4.17	4	3.67	4.17
Usability	4.83	4.17	4.67	4.33

Average Rating 1 – 5 (strongly disagree – strongly agree)	
Question No.	15
	3.33

Comments:

- "Well defined measures and relevant to business of BT".
- "Automated methods exist for customer feedback (e.g. complaint prediction based on their interaction and their service performance)".
- "Innovative".
- "Might lead to some inaccuracy".
- "Virtually all business process reengineering techniques that I have seen in the past rely significantly on customers and customer related data".
- "The only criteria to consider when providing certain services"

9.2.6. Data

18) Database structure (e.g. reflects taxonomy structure, enables cost estimating framework).

19) Information content.

Average Rating 1 – 5 (poor – excellent)		
Question No.	18	19
Effectiveness	4.67	4.5
Accuracy	5	4.17
Usability	5	4.33

Comments:

- "Straight forward and familiar to business users".

9.2.7. Integrated Technique

20) Meets its aim – to develop a framework to automate business process reengineering using cost, duration and impact on customer satisfaction as criteria.

21) Achievement of objectives and implementation:

   a) Development of an automated reengineering plus optimisation approach for business processes using cost, duration and customer satisfaction criteria.
   b) Development of a mechanism for representing a process model to allow input to an algorithm.
   c) Development of an operational cost modelling approach for business processes and a means of detecting impact on customer satisfaction.
   d) Development of a prototype software platform to validate the approach.
   e) Once implemented, with periodic maintenance can always have optimum available processes in operation.
   f) Implementation issues no worse than any other new system implementation in large companies.
   g) Can be applied to any industry – not only the service sector.

22) Time and Problem Complexity:

- The example ‘Place Telephone Order’ finds the optimum process in average 20 seconds.
- The generic example finds the optimum process in average 85 seconds.

   a) It is proposed that this technique is significantly faster than a human/manual attempt at reengineering.
   b) Given that the database for a large organisation will contain very large amounts of process data, it is proposed that a human/manual attempt at reengineering may be impossible (e.g generic process example with

129,140,163 possible combinations, each step varying in cost and duration).

Average Rating 1 – 5 (strongly disagree – strongly agree)										
Question No.	20	21a	21b	21c	21d	21e	21f	21g	22a	22b
	5	4.67	5	4.83	4.33	4.17	4	4.17	5	5

Comments:

- "Very important for service industry and could have impact on future SOA-based models".
- "Quite impressive".
- "Fine tuning required".
- "Strong in converting graphical models to coded representations".
- "Java or .Net would have been better implementation platforms".
- "Not sure if applies to other sectors without modification".

9.3. Analysis of Results

The test scenarios demonstrated the automated reengineering approach and how it can apply some of the guidelines of BPR at a process level. Each of these guidelines was applied, either individually or in combination, leading to reengineered processes. The optimisation stage of the technique, the GA based module, was also demonstrated, and its value and effectiveness proved through comparisons based on numbers of alternative tasks available and process length.

The integrated BPR technique, along with its component parts, was generally well regarded by all interviewees, and well received by the industrial sponsors. The overall average rating was 4.4 on a scale of 1 to 5. Some weaknesses, along with strengths, were pointed out, which were accepted by the researcher. Ratings, comments and opinions are discussed in Sections 9.3.1 to 9.3.6.

9.3.1. Process Coding Mechanism

A strength pointed out with the taxonomy creation was that it is one of the more effective and straightforward ways in which to develop a hierarchal structure, but as with any application that leads to the development of a hierarchal structure, conflicts can arise and will need to be managed. As a result, the average rating for the taxonomy was **4.17** (Question 1), with all ratings either **4** or **5**.

A weakness highlighted with the modelling technique was its usability in the creation of the process codes – ideally there should be an automated translation mechanism to transform the diagrammatic model into a coded model, but it was accepted that a good user interface design would assist with this task considerably. This led to a lowest rating of **3** for the coding mechanism (Question 2), average **3.67**, in the ***usability*** category and **2** for the recommended translation to BPMN flow diagrams (Question 3), average **3**, also in the ***usability*** category. A strength pointed out was that this technique makes graphical models amenable to processing with algorithms such as GA, which was the intention of the research, and is very strong in converting graphical models into coded representations.

Due to the strengths of the coding mechanism that were highlighted, leading to a highest rating of **5**, the average overall rating for the process coding mechanism was **4** (Questions 1 to 3), with averages of **4.22** for ***effectiveness***, **4.17** for ***accuracy***, and falling to **3.61** for ***usability*** due to concerns about ease of translation into coded representations.

9.3.2. Automated Reengineering

The strength of this technique is that it provides comprehensive reengineering of processes, as intended, but could involve continuous or at least periodic maintenance.
A weakness is that it may not be able to fully encapsulate the subjectivity in BPR (the lack of human judgement), thus leading to a lowest rating of **3** (Question 4) in both ***accuracy*** and ***usability***, averages **3.5** and **3.67** respectively.

It was pointed out that the GA based section is a very usable part of the methodology as it does not require intimate knowledge of GA by the user. For instance, analysing a pareto front generated by a MOO would require this knowledge, but this methodology produces results that are tangible for someone from a business/managerial background. As such, a high rating of **5** was achieved for the GA based section, with a unanimous agreement and average of **5** on Question 5a (maximisation of search space), and an average of **4.61** on Question 5b, giving an average rating of **4.81** for the GA based section. The examples provided were accepted as realistic and explanatory (Question 6), giving an average of **4.17**, and unanimous agreement was again achieved on Question 7, the superiority of output processes, giving an average of **5**.

The average overall rating for automated reengineering was **4.37** with averages of **4.34** for ***effectiveness***, **4** for ***accuracy***, and **4.25** for ***usability*** (Questions 4 and 5b).

9.3.3. Cost Estimation

The ABC structure and methodology was perceived as effective and clear to business and financial systems users by most respondents, but one respondent indicated that, without an adequately qualified cost estimator (or statistical analyst), some users of this information may be unable to fully utilise it due to a lack of understanding**.** It was also pointed out that the technique does rely on execution data for duration and probabilities, which may not always be available.

Due to a large number of **4** and **5** ratings in Questions 9, 10 and 12, the overall average rating of cost estimation was **4.51**, with averages of **4.73** for ***effectiveness***, **4.67** for ***accuracy*** and **4.14** for ***usability***. Low ratings of **3** were given for usability of the ABC structure (Question 8), average rating **3.67**, and the automated method of estimating process cost (Question 11), average rating also **3.67** – this was due to the potential problem of understanding by some users and reliance on process execution data, as mentioned above.

9.3.4. Customer Satisfaction

The ‘Two Times’ criteria (Question 13) were regarded as well defined measures which are relevant to the industrial sponsor’s needs, leading to an average rating of **4.61**. The formula used, based on available information, and the use of the value generated (Questions 14, 16 and 17) was described as innovative but may lead to some inaccuracy. This led to low scores of **3** in ***accuracy*** of the formula, with an average of **3.67**. Overall averages for these three questions were **4.39** for ***effectiveness***, **3.95** for ***accuracy*** and **4.39** for ***usability***.

Also, it was pointed out that automated methods exist for customer feedback, such as complaint prediction based on their interaction and service performance – but this does, however, rely on analysis of customer related data, which is not available with this methodology. Another respondent differed, and strongly agreed that this was in fact the only method available without customer data. Another pointed out that, dependant on the service provided, the ‘Two Times’ may be the only criteria to consider. Thus, the average rating for Question 15 was **4**, which included a lowest rating of **3** and highest of **5**.

Therefore, despite some concerns over accuracy, the overall average rating of customer satisfaction was **4.31**, with averages of **4.5** for ***effectiveness***, **4** for ***accuracy*** and **4.5** for ***usability*** (Questions 13, 14, 16 and 17)

9.3.5. Data

The database, as intended, was accepted as straight forward and familiar to business users. As such, ratings consisted only of **4**s and **5**s (Questions 18 and 19), leading to an overall average of **4.62**, with averages of **4.59** for ***effectiveness***, also **4.59** for ***accuracy***, and **4.67** for ***usability***.

9.3.6. Integrated Technique

It was unanimously agreed that the research meets its aim (Question 20), with an average rating of **5**. It was classed as very important for the service industry and could have an impact on future service oriented architecture (SOA) based models. It was also agreed or strongly agreed (ratings of either **4** or **5**) that the research meets all of the required objectives (Questions 21a to 21d), producing an average rating of **4.71**.

It was considered that fine tuning might be required for the integrated technique – to be expected, as all organisations operate differently. It was also considered that the integrated technique may need some modification dependant on the industrial application (i.e. other than the service sector – Question 21g). However, it was designed with flexibility and scalability in mind, and so this would not necessarily present a significant problem – thus producing an average rating of **4.17**.

The prototype software platform (Question 21d), ABPRS, was accepted as meeting its requirements, but the industrial sponsor commented that they would have preferred Java or .Net as implementation platforms. However, the platform could be transferred to another platform if required – it was developed in a structured, modular fashion, and all VB code is heavily commented - which would facilitate this task. As a result, the average rating for ABPRS was **4.33**.

As far as implementation of the system is concerned (Question 21f), it was generally agreed or strongly agreed that implementation would not be more difficult than other systems implementations within large organisations, thus generating an average rating of **4.17**, which included one rating of **3**. In terms of maintenance following implementation, and resulting process optimality (Question 21e), it was unanimously agreed that this would be the case, with ratings of **4** from each respondent.

Finally, the purpose of attempting to automate BPR as far as possible, the saving of the human time and effort involved (Questions 22a and 22b), generated unanimous strong agreement (all ratings of **5**) – significant time would definitely be saved and the system could potentially achieve reengineering which may not be possible via human effort due to time and problem complexity.

.

9.4. Summary

This chapter provided a number of test scenarios in order to demonstrate the functionality of the integrated reengineering approach, and presented the results of expert opinion on the research, giving ratings and comments on all aspects of the research. Some of the comments made are discussed further in the next chapter.

THIS PAGE IS INTENTIONALLY LEFT BLANK

10. Discussion and Conclusions

This chapter presents the achievement of the aim and objectives of the research, the contributions made, the limitations of the research, the impact on businesses of implementing the results of the research, and further work that may be undertaken in the future.

10.1. Summary of the Achievement of Aim and Objectives

The research aim was to develop a framework to automate business process reengineering using process cost, duration and impact on customer satisfaction as criteria. In order to realise the aim of the research, a number of objectives were set in the initial stages of the project (Chapter 3), derived from the research gaps identified in Chapter 2, which then became a basis for the research to follow. The research has successfully achieved all of these objectives - this section discusses the achievement of each of them.

10.1.1. Achievement of Objective 1

Objective 1 was to develop a mechanism for representing a process model to allow input to an algorithm. This was achieved firstly through review of literature, as presented in Chapter 2. The review was reinforced by the case studies with the industrial sponsor in which process models were obtained and analysed. In order to accurately target the research, a number of criteria were set, as follows:

- The process representation should be readable and understandable without specialised knowledge, such as computer science or mathematics.
- It should be possible for a user without specialised knowledge to amend or create process models (with the assistance of a GUI).
- It should allow quantitative (measurable) analysis rather than purely qualitative analysis – it is not possible to examine and reengineer a process via algorithmic means without quantitative information (for example, the duration of a task).

Therefore, it should be represented with an appropriate level of detail to allow this.

- It should allow representation of complex processes - service sector business processes are based on human to human or human to machine (i.e. web-based processes) interaction and, as a result, may involve branches and/or repetitions based on decisions and events.
- The model should contain sufficient information to analyse each element of the process, individually or in sections, and compare with other process data held in a data warehouse, in order to potentially reengineer to the extent where complete process reconfiguration could occur.
- The detailed (dynamic and functional), quantitative process model should allow storage in a database and retrieval for purposes of analysis (it should be possible to store the model and then retrieve it in exactly the same state as it was stored in).
- It should be possible to link cost and duration measures to the steps in the model.

All of these criteria were considered during the research, and it became apparent that no existing modelling tools and techniques could cater for all of them. Ideas were generated by the review of process modelling techniques, notably the flow diagram due to its ability to depict process flow at the desired level of detail, along with WFMS and the facility to transform diagrammatic representations into computer code in order to enable execution of the process. Further research into group technology, classification and coding systems, and taxonomies led to the creation of the representation methodology. This differs from other representation techniques in that it is a purely numeric representation – it is not diagrammatical or mathematical, it is simply a string of codes, one for each element of the process, which, when strung together, creates a comprehensive numerical process model containing details of all inputs, outputs, constraints, resources used, type of element and flows in and out, next/previous steps, and a unique identifier for each element based on the creation of a company taxonomy. The process representation methodology is presented in Chapter 4.

10.1.2. Achievement of Objective 2

Objective 2 was to develop an operational cost modelling approach for business processes. This was achieved through review of literature (Chapter 2), a cost estimating survey and an online forum (Chapter 6). Literature and the companies involved in the survey indicated the direct link between activities and business processes, and therefore the suitability of ABC. Along with the survey and forum, it became apparent that business processes may follow numerous different paths due to the outcome of decisions or events, and that, when reengineering processes such as this, the probabilities of certain paths being followed may change. This situation should have been catered for when initially allocating costs to activities and during periodic updates, but when probability of resource usage changes during automated reengineering, this effect on cost should be immediately recognised and the process cost re-estimated. This was the principal reason behind the proposed ABC methodology. The survey highlighted a degree of flexibility in cost estimating methodologies, generally conducted at a relatively high level (e.g. departmental), and it was therefore proposed to add another level of detail to a typical ABC structure by introducing a direct resource to task relationship – this allows differentiation principally between reengineered processes that use the same resources but in different degrees due to changes in probabilities of a process following certain routes. The ABC methodology is presented in Chapter 6.

10.1.3. Achievement of Objective 3

Objective 3 was to develop a means of detecting impact on customer satisfaction. This was achieved through review of literature (Chapter 2) and interviews. In order to measure customer satisfaction, the customer must either provide feedback, or data on his/her behaviour must be collected and analysed – perhaps via a Customer Relationship Management (CRM) system. A methodology for measuring impact on customer satisfaction was required that did not use customer related data – during automated reengineering, it would not be possible to use such data. Therefore, as a result of literature review and interviews, it became clear that the available relevant information in the process model, duration and gateway count, could be used to provide an approximate measurement of impact – notably, the ‘Two Times’. Chapter 7 describes the methodology developed.

10.1.4. Achievement of Objective 4

Objective 4 was to develop an automated reengineering plus optimisation approach for business processes using cost, duration and customer satisfaction criteria. The automated reengineering methodology was developed mainly as a result of literature review (Chapter 2), case studies, workshop and iterative experimentation. The desire was to take the coded process model and apply to it the generally accepted guidelines for the reengineering of business processes at this level, normally undertaken by human experts facilitated by IT. This was due to the potentially very large numbers of possible reengineered processes that could be generated by a large organisation running many service based processes, and with many available alternative steps for those processes – for instance, a multitude of external service providers (e.g. suppliers, couriers) and available technologies. In conjunction with the steps of reengineering, a GA based optimisation module would maximise the available search space, and generate and analyse potentially all possible feasible processes available – necessary when dealing with large organisations’ data. With vast numbers of reengineered processes which may produce the same output, but which are comprised of activities with tasks which all have differing costs and durations, the human effort involved in finding the optimum process would be considerable, if not impossible. Therefore, using the developed coding mechanism as an input, the cost estimating methodology as the principal means of evaluation, along with duration and the customer satisfaction methodology, the automated reengineering methodology was developed, as presented in Chapter 5.

10.1.5. Achievement of Objective 5

Objective 5 was to develop a prototype software platform. The development of the platform, ABPRS, was conducted in conjunction with the integrated reengineering methodology. Literature review assisted in the database design, and the first case study with the industrial sponsors provided a variety of technologies which were tested and experimented with. The platform is self contained and was developed in a structured and modular fashion. It was intended that it should be flexible and scalable for future customisation, or so that one or more of the component methodologies can be extracted and used independently. Chapter 5 describes the methodology which it executes, and Chapter 8 gives further details of its appearance and functionality.

10.1.6. Achievement of Objective 6

Objective 6 was to evaluate the integrated automated business process reengineering technique. This was achieved via the prototype software platform in which a number of test scenarios were created and executed, producing the desired result in all cases. The research was also presented to six experts who provided their expert opinion on all aspects of the integrated methodology. Chapter 9 provides full details of this.

10.1.7. Achievement of Research Aim

By achieving the objectives described above, the overall aim of the research was achieved: to develop a framework to automate business process reengineering in the service sector using process cost, duration and impact on customer satisfaction as criteria. The opinions of a number of experts, which were generally favourable (as presented in Chapter 9), reinforce this achievement.

10.2. Research Contributions

Through achievement of the aim and objectives of the research, a number of contributions to knowledge have been made, as follows:

1. An automated reengineering technique which attempts to follow the guidelines of BPR as closely as possible at a process model level. It has been developed as a hybrid algorithm, using both conventional and evolutionary (genetic algorithm) computing, and allows automated input, analysis, evaluation and reengineering of complex process models based on cost, duration and impact on customer satisfaction, and which also enables reconfiguration based on these criteria. This involves the automated combination of business process reengineering and

optimisation, ABC and evolutionary computing, and its application to the service sector.

2. The development of a mechanism for process representation which enables comprehensive modelling of complex processes via a string of numeric codes for input to algorithms. This also allows comparisons to be made with alternate process data with a view to replacing steps or sections of existing processes.

3. An ABC framework which allows for an extra level of detail through the inclusion of direct task-resource costs as well as the allocated activity centre costs. The ABC methodology developed within this framework is probability driven and principally enables differentiation between processes which may use exactly the same resources, but in differing degrees based on the probability of certain routes in the process being executed.

4. A customer satisfaction change detection methodology which does not rely on customer feedback or customer related data to evaluate potential impact, both negative and positive, of reengineering of processes.

The above contributions are illustrated in the following diagram, Figure 98, which shows the elements of the developed system.

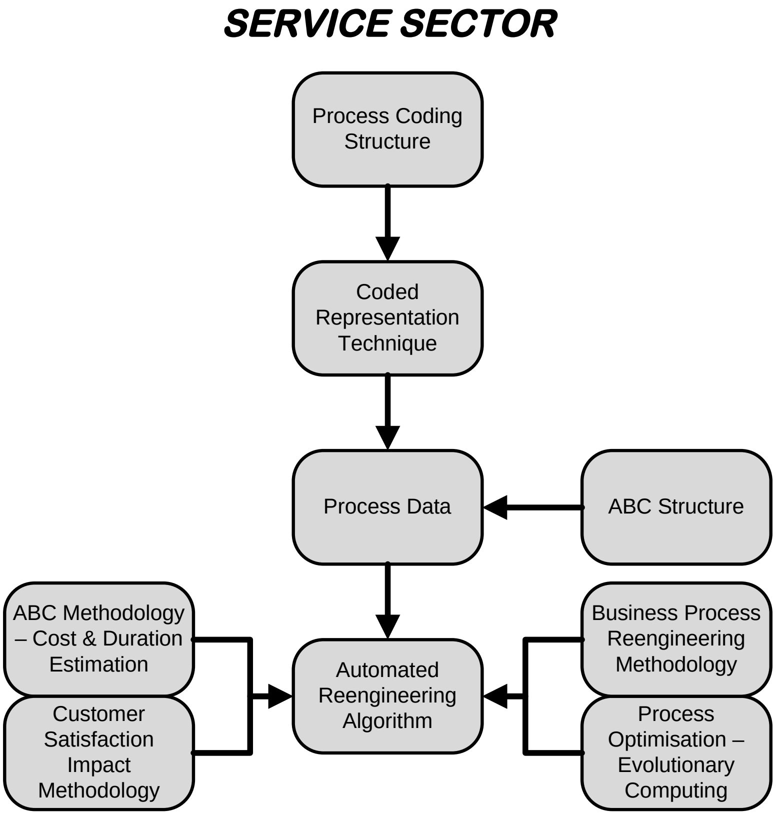

SERVICE SECTOR
Process Coding Structure
Coded Representation Technique
Process Data
ABC Structure
ABC Methodology – Cost & Duration Estimation
Customer Satisfaction Impact Methodology
Automated Reengineering Algorithm
Business Process Reengineering Methodology
Process Optimisation – Evolutionary Computing

Figure 98: Final Contributions Summary

10.3. Limitations

A number of limitations became apparent through the development and validation stages of the research. These are discussed below with suggestions for improvement or alleviation.

10.3.1. Process Coding Structure and Methodology

A limitation highlighted with the modelling technique was its usability in the creation of the process codes. Ideally, there should be an automated translation mechanism to transform the diagrammatic model into a coded model, but it was accepted that a good user interface design would assist with this task considerably. This would basically involve pick lists, providing details of resource centres with all allocated input and output codes, with the option of creating new codes. Flows and step types would be selected from available BPMN notations, resource codes from those already in use with the option of inputting new ones, and constraints from those available which already cater for every eventuality.

Similarly, manually translating the reengineered and optimised process report produced back into a BPMN model may be considered a limitation. Although this is not a difficult task, it can be time consuming.

Another aspect that may be considered a limitation is the proposed implementation structure (i.e. taxonomy creation leading to unique step identifiers). However, this can be bypassed and a process model translated either into a flow diagram, assuming it is not already represented in this way, and then into a coded representation, or directly into the coded representation. Translation into a BPMN flow diagram was successfully achieved using an IDEF3 and an ARIS model (see Chapter 4). The unique identifier could then be randomly generated, thus losing the descriptive nature of the identifier, but removing the work involved in creating the company taxonomy and its codes.

10.3.2. Automated Reengineering Methodology

This is limited mainly due to the fact that it is unable to use the human expert judgement normally applied in BPR, due to automation. It also can only use existing process data to reengineer processes – it cannot create new process steps. However, as all process alternatives generated have differing costs and durations, the human effort involved in finding and maintaining the optimum process is considerable, especially when potentially millions of processes can be created from the existing process plus a selection of possible alternatives. When a company is running hundreds or thousands of different processes, this situation is further amplified.

10.3.3. ABC Structure and Methodology

The ABC methodology relies on correct duration and probability data in order to produce accurate estimates. Therefore, process execution data must be available in order to assign duration to process tasks and probabilities to gateways. Also, the duration and probabilities must be periodically updated in order to maintain accuracy. This requires consideration as does the structure of the ABC implementation which must incorporate a direct resource to task relationship. Without this relationship and the duration and probability data, the methodology cannot work effectively. The database allows for the inclusion of a process execution log to facilitate updates.

10.3.4. Customer Satisfaction Methodology

Although the majority of customers appreciate a swift, efficient and uncomplicated process, there will always be a small minority who may appreciate spending a longer time on the phone or the internet – reducing cycle time may have a negative impact for these customers and their perceived quality of the service provided detracted from. Furthermore, decreasing the duration of a process may detract from the actual, rather than perceived, quality of the service provided – for instance, the removal of a customer courtesy related activity or task which would reduce the duration (and therefore cost) of a process, but still enable the same output to be generated, could have an adverse impact on levels of satisfaction with the process. However, this situation can be avoided by identifying such high value activities or tasks and setting constraints accordingly (e.g. ‘Must be Present’) in order to prevent their removal from the process by ABPRS. Also, the removal of gateways may not necessarily have a positive impact on satisfaction levels: some gateways will allow a customer to quickly talk to the correct people or arrive at the correct web page and each flow out of the gateway will be equally beneficial - removal of this type of gateway may risk creating a negative impact. In general though, the technique provides as accurate an indication as possible with the information available within the elements of a business process model, in a situation where retrospective feedback or direct analysis of customer related data is not an available option.

10.3.5. Integrated Approach - ABPRS

As with any new computer system (e.g. ERP or WFMS), there are implementation issues associated with this technique but, once a company’s process models are assigned the necessary numerical codes and stored in the RDB, which is quite a time consuming process, it is simply a matter of periodic maintenance (i.e. when a new process step becomes available, it is coded and stored in the system, and the optimisation process is executed). The issues of implementation are also presented in the next section, 10.4.

10.4. Business Impact Analysis

The impact of implementing, rolling out and then maintaining a system such as the one described in this thesis is not inconsiderable. However, when compared to the ‘fundamental’ and ‘radical’ changes involved in BPR at an enterprise-wide level, which may involve complete organisational restructuring, the impact is much less significant and disruptive. The system can be implemented as a background task and be essentially ‘invisible’ to the majority of an organisation – it should not interfere with most of a company’s everyday operations. The principal factors involved are as follows:

10.4.1. Business Process Data

All company processes require translation into coded representations in order to create a process library such as the sample library presented in Chapter 4. This could potentially be achieved via one of the following procedures:

1. Create a company taxonomy
   Create BPMN flow diagrams from existing process models
   Create coded process representations from BPMN flow diagrams

2. Create a company taxonomy
   Create coded process representations from existing process models

3. Create BPMN flow diagrams from existing process models
   Create coded process representations from BPMN flow diagrams

4. Create coded process representations from existing process models

The fastest, but not necessarily most effective option is number 4, but the option selected is dependant on the individual company’s situation – for instance, existing company structure, time and resources available, modelling expertise available, format of existing process models, and the desire to follow recommended implementation structure. This is certainly a time-consuming process which involves a degree of expertise in process architecture. The resultant process library will be extensive and contain a very large amount of process data – every one of the process steps, sub-processes and complete processes that the organisation uses must be encoded and stored in the company's RDB. It is envisaged that this task will be undertaken by process owners and/or process architects.

Cost, duration and probabilities data are required in order for the system to function. Depending on the available company resources, this can be extracted from sources such as OLTP and/or WFMS/BPMS systems, ABC breakdowns, or more basic process logs recorded by process owners. This will also involve adjustment of overheads cost allocation per activity centre to allow for the direct resource costs attached to process tasks which are used in cost estimates (as described in Chapter 6) by the system – the aggregation of the direct resource costs and adjusted overheads per activity centre should equate to the overheads prior to adjustment.

10.4.2. Maintenance

As potential alternatives become available (e.g. new tools and technologies, new external service providers, etc.), these require encoding and entering into the system, either as single process steps or sub-processes.
Cost, duration and probabilities data should be updated as frequently as possible, but monthly would be reasonable.

10.4.3. Required Resources

In order to create and maintain the process database, including cost, duration and probabilities data, it is recommended that an expert in process architecture and management takes overall responsibility for the system, along with change management responsibilities. Process owners should also be involved in this. Training should be provided for those employees with lesser experience in process modelling and management who are to be involved in the project, as further work (see Section 10.5) will facilitate the use and maintenance of the system. It is envisaged that, in the large organisations that the system is designed for, these employee resources will be available internally. However, it may be necessary to hire external skills for a period of time if this is not the case.

It will also be necessary, particularly in the implementation stage, to involve the organisation’s accountants and/or cost engineers to supply and adjust cost data for the system.

10.4.4. System Usage

Once implemented effectively and after a period of testing, ABPRS should be executed for every process in the organisation by the employees who execute or manage processes (i.e. process owners). The resultant reengineered process models should then be reviewed by process architects and/or process owners, not only to ascertain their validity (they may be unsuitable in some cases due to factors not catered for within ABPRS), but also to ascertain the extent of the changes proposed. It is likely that a large number of the organisation’s processes may remain completely or almost unchanged, but some will include significant changes. Once the initial results have been checked, a strategy for implementing the proposed changes should be decided on – it may be advisable to stagger the implementation over a period of time. In cases of significant impact due to reengineering, an adjustment period may be required by company employees before the next set of changes are implemented, depending on the circumstances and nature of the impact.

Once all initial process improvements have been implemented, the system should then be run periodically for every process – preferably each time cost, duration and probabilities data is updated. It is feasible that what was identified as the optimum process previously may now no longer be the optimum due to changes in resource usage and duration, perhaps as a result of factors such as price increases of external service providers, lack of availability of external products or services, problems with the products or services the company provides, new staff, and software problems. As such, the duties of process owners may be adjusted periodically as ABPRS reengineers processes according to these changes. .

Finally, it is likely that the recipients of reengineered processes, the customers, may also be affected by the implementation and usage of ABPRS - but the intent is that this effect is a positive one which will beneficial not only to the customer, but also to the organisation.

Potential further work which will enhance and facilitate the use and maintenance of the system is described in the next section.

10.5. Further Work

This relates to the limitations presented in Section 10.3 and the business impact described in the previous section.

- A more sophisticated graphic user interface can be incorporated into the software platform to facilitate process coding.

- Also, the system could be modified to import process representations – for instance, JBoss jBPM process representations are held in XML format – this code could partially be translated into the numerical representations presented in this research.

- A facility to translate the output reengineered process coded representation back into a BPMN flow diagram

- The system supports a process execution log for analysis of past process runs. When data is supplied, this will allow intermittent updating of duration, cost and probabilities, thus allowing more accurate estimates of the cost of a process.

The previous points relate to the software platform and how it can be extended. However, in terms of the integrated methodologies presented in this thesis, future work and research could potentially include the following:

- Integration of one or all of the methodologies with ERP systems, such as SAP.

- Integration of the automated reengineering technique with WFMS or BPMS. The coding mechanism could potentially be used as a replacement for WFMS/BPMS process languages, perhaps with a translation facility incorporated to achieve this, or a link between ABPRS and WFMS/BPMS to enable importing of process representations (as suggested in Section 10.5), and then partial or complete translation to coded process representations attempted. Also, the integration or linking could allow automatic updating of cost, duration and probabilities data.

- Use of the customer satisfaction impact methodology in CRM systems or in CRM modules of ERP systems.

- Further work on the GA part of the technique to include:

  - Multi-objective optimisation (via the addition of more objectives).
  - Use of GA to reconfigure process models – earlier attempts in this research had limited success due to the unconstrained, free-standing nature of many process steps (see Chapter 5). Further research may improve on this.

10.6. Conclusion

The main findings of this research were as follows:

- Process models are mainly diagrammatical in form in order to facilitate ease of understanding. This makes them qualitative in nature and impossible to input to or analyse quantitatively by algorithmic means. Formulaic or mathematical process representations exist which allow input and quantitative analysis, but they cannot represent complex processes, generally catering for serial processes with very few steps. Programmed representations also exist, used by WFMS and BPMS to enable execution of processes, but comprehensive process representation is an issue, along with sufficient information being available to allow comparisons of the process model with potential alternative process data.

- ABC structures are implemented with a degree of flexibility according to organisational requirements, but they share many similarities and follow the basic principles of ABC. However, ABC does not cater for estimating the cost of processes whose component steps may vary in their amount of resource usage due to the outcome of decisions or events during the process execution - this should have been catered for when originally assigning costs to activities and during periodic updates, perhaps based on online transaction processing data, but if a process is to be reengineered and this resource usage would change as a result, it must be instantly detected and measured as it will effect the cost of the process. Cost data is generally collected at a relatively high level, such as departmental, and so changes in resource usage are indirectly detected, but only when this data is collected and implemented in the cost estimating system. Also, ABC does not assign costs at task level, but purely at activity or activity centre level, making it impossible to instantly differentiate between processes which may vary in the direct resources they use.

- Customer satisfaction is generally measured and monitored via customer feedback or analysis of customer behaviour – the most effective methods for compiling customer data. Situations are not catered for in which an instant judgement is required on a reengineered process’s impact on satisfaction without the availability of customer feedback or process related customer data.

- BPR is carried out at a number of different levels, generally undertaken by human experts (with the assistance of IT) capable of using their judgement. In large organisations with data warehouses containing vast amounts of process data, the human effort in maintaining complete process optimality is considerable, if not impossible.

As a result of the above findings, research gaps were identified and the integrated BPR methodology was developed in an attempt to fill these gaps. The research demonstrates the following:

- It is possible to represent a business process model as a string of digits via a coding mechanism.

- The coded representation can be analysed and manipulated by an algorithm.

- It is possible to extend the automation of BPR at a process model level a step further than other techniques in an attempt to replace human expert judgement, thus enabling infinitely faster reengineering of processes to take place.

- Probabilities can be included in an activity based cost estimate in an attempt to add another level of detail, and therefore accuracy, to the estimate.

- Impact on customer satisfaction may be measured without the use of customer related data.

This chapter has reiterated the aim and objectives of the research and how they were achieved. It has described the findings of the research and the contributions that stemmed from these findings. Limitations and future work have been identified, which became apparent either during the development phases or through expert opinion. The impact on businesses of implementing and using the developed system has also been discussed.

The research presented in this thesis led to the creation of a number of methodologies that were integrated. The component methodologies, i.e. the process coding mechanism, the ABC methodology and the customer satisfaction impact methodology, can also be used independently, perhaps becoming integral parts of other systems, as suggested above. The methodologies, either independently or in integrated form, can also be applied to domains other than the service sector. This may require some degree of customisation, but the approach has been developed with flexibility in mind – for example, as long as a process model exists and is coded according to the methodology described in this research, even if this should require extensions to the process codes (i.e. certain processes in other domains, such as Manufacturing, may require slightly different or extra information to enable comprehensive representation) detailed in this thesis, it can be input to the automated BPR system, regardless of its application domain, and potentially be reengineered.

It is envisaged that, once implemented effectively, the automated BPR system could cause immediate and perhaps ‘radical’ improvements within an organisation in terms of cost and levels of customer satisfaction. By regularly updating cost, duration and probabilities data, the optimum processes can always be maintained, thus providing the facility to automatically reengineer business processes which instantly adapt in response to a constantly changing market – the optimum process one week may not be the optimum the following week.

Furthermore, apart from the significant benefits to an organisation that the system may provide in terms of reengineering processes to reduce cost and increase levels of customer satisfaction, another benefit may be that more time is available to personnel who previously were involved in the manual, iterative procedure that the automated system attempts to replace – thus allowing these employees to concentrate on more innovative tasks which, in turn, may further enhance the benefits to an organisation.

THIS PAGE IS INTENTIONALLY LEFT BLANK

References

12MANAGE *Activity Based Costing Forum* (2008), available at: www.12manage.com (accessed 2008).

AACE (1990), *Standard Cost Engineering Terminology. AACE Recommended Practices and Standards.*

Aderobal, A. (1997), 'A Generalised Cost Estimation Model for Job Shops', *International Journal of Production Economics* , Vol. 53, No. 3, pp. 257-263.

Aguilar-Saven, R.S. (2004), 'Business Process Modelling: Review and Framework', *International Journal of Production Economics,* Vol. 90, No. 2, pp. 129-149.

Aliga, T. and Sarmenta, R. (2001), 'Activity Based Budgeting', *SGV Bulletin,* Vol. 5, No. 5, pp. 61-69.

Baeyens, T. (2004), *The State of Workflow*, available at: http://www.theserverside.com/tt/articles/article.tss?l=Workflow (accessed 2008).

Baker, V. (2001), *Full Supplier Cost Modelling (MSc Thesis)*, Cranfield University, UK.

Band, W.A. (1991), *Creating Value for Customers: Designing & Implementing a Total Corporate Strategy*, Wiley, New York.

Bashir, H.A. and Thomson, V. (2001), 'Models for Estimating Design Effort and Time', *Design Studies,* Vol. 22, No. 2, pp. 141-155.

Bavan, A.S. and Abeysinghe, G. (2005), 'On the Specification and Implementation of Distributed Systems Using NMDS and LIPS', *Information and Software Technology,* Vol. 47, No. 11, pp. 721-734.

Bernus, P. and Nemes, L. (1996), *Modelling and Methodologies for Enterprise Integration*, Chapman & Hall, London.

Bradley, P., Browne, J., Jackson, S. and Jagdev, H. (1995), 'Business Process Re-Engineering (BPR) - a Study of the Software Tools Currently Available', *Computers in Industry*, Vol. 25, No. 3, pp. 309-330.

Brimson, J.A. (1998), 'Feature Costing: Beyond ABC', *Journal of Cost Management*, Vol. 4, pp. 6-12.

Burke, G. and Peppard, J. (1994), *Examining Business Process Re-Engineering: Current Perspectives and Research Directions*, Kogan Page, London.

Burns, R.B. (2000), *Introduction to Research Methods*, Sage Publications Ltd., London.

Börjesson, S. (1994), 'What Kind of Activity-Based Information Does Your Purpose Require?', *International Journal of Operations & Production Management,* Vol. 14, No. 12, pp. 79-100.

Cameron, N.S. and Braiden, P.M. (2004), 'Using Business Process Re-Engineering for the Development of Production Efficiency in Companies Making Engineered to Order Products', *International Journal of Production Economics*, Vol. 89, No. 3, pp. 261-273.

Carr, C. (1990), *Front-Line Customer Service: 15 Keys to Customer Satisfaction*, Wiley, New York.

Chalos, P. (1991), *Managing Cost in Today's Manufacturing Environment*, Prentice-Hall, New Jersey.

Chen, P.P. (1976), 'The Entity Relationship Model - Toward a Unified View of Data', *ACM Trans. Database Systems 1*, Vol. 1, No. 1, pp. 9-36.

Choudrie, J., Hlupic, V. and Zahir, I. (2002), 'Teams and Their Motivation for Business Process Reengineering: A Research Note', *The International Journal of Flexible Manufacturing Systems*, Vol. 14, No. 6, pp. 99-107.

Codd, E.F. (1990), *The Relational Model for Database Management*, Addison-Wesley, Reading, MA.

Coello Coello, C.A., Van Veldhuizen, D.A. and Lamont, G.B. (2002), *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, New York.

Cokins, G. (1996), *Activity-Based Cost Management Making It Work*, McGraw Hill, Boston.

Cokins, G. (1998), 'ABC Can Spell a Simpler, Coherent View of Costs', *Computing Canada*, Vol. 24, No. 32, pp. 34.

Cokins, G. (2001), *Activity-Based Cost Management An Executives Guide*, Wiley, New York.

Date, C.J. (1990), *An Introduction to Database Systems, Vol. 1*, Addison-Wesley, Reading, MA.

Davenport, T.H. and Short, J.E. (1990), 'The New Industrial Engineering: Information Technology and Business Process Redesign', *Sloan Management Review,* Summer 1990, pp. 11-27.

Dean, E.B. (2003), *Activity Based Costing Resources*, available at: www.offtech.com.au/abc/ABC_Resources10.asp (accessed 2005).

Deb, K. (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester.

del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.-C. and Harley, R.G. (2008), 'Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems', *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 2, pp. 171-195.

Denzin, N.K. and Lincoln, Y.S. (1998), *Collecting and Interpreting Qualitative Materials*, Sage Publications, California.

Dickinson, V. and Lere, J.C. (2003), 'Problems Evaluating Sales Representative Performance? Try Activity-Based Costing', *Industrial Marketing Management,* Vol. 32, No. 4, pp. 301-307.

Duxbury, D., Backhouse, R., Head, M., Lloyd, G. and Pilkington, J. (1999), 'Call Centres in BT UK Customer Service', *British Telecommunications Engineering,* Vol. 18, No. 3, pp. 165-173.

Engineering Village (2008) available at: www.engineeringvillage.org (accessed 2008).

Flick, U. (2002), *An Introduction to Qualitative Research*, Sage Publications Ltd., London.

Fornell, C. (1992), 'A National Customer Satisfaction Barometer: The Swedish Experience ', *Journal of Marketing,* Vol. 56, No. 1, pp. 6-21.

Gallagher, C.C. and Knight, W.A. (1986), *Group Technology Production Methods in Manufacture,* Ellis Horwood Ltd, New York.

Gen, M. and Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, John Wiley, Chichester, UK.

Gerson, R.F. (1993), *Measuring Customer Satisfaction*, Kogan Page, London.

Gill, J. and Johnson, P. (2002), *Research Methods for Managers*, Paul Chapman, London.

Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, Reading, MA.

Gordijn, J., Akkermans, J.M., and van Vliet, J.C. (2000), 'What's in an Electronic Business Model', *12th International Conference on Knowledge Engineering and Knowledge Management EKAW-2000, 2-6 October 2000*, Juan-les-Prins.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M. and Shan, M.-C. (2004), 'Business Process Intelligence', *Computers in Industry,* No. 53, pp. 321-343.

Grover, V. and Kettinger, W.J. (2000), *Process Think: Winning Perspectives for Business Change in the Information Age*, Idea Group Inc, Hershey, PA, USA.

Grover, V. and Malhotra, M.K. (1997), 'Business Process Reengineering: A Tutorial on the Concept, Evolution, Method, Technology and Application', *Journal of Operations Management*, Vol. 15, No. 3, pp. 193-213.

Gummesson, E. (1991), *Qualitative Methods in Management Research*, Sage, London.

Gunasekaran, A. and Kobu, B. (2002), 'Modelling and Analysis of Business Process Reengineering', *International Journal of Production Research,* Vol. 40, No. 11, pp. 2521-2546.

Hackney, J.W. (1965), *Control and Management of Capital Projects: Dynamic Estimating, Control, and Management by Owners Corporations of the Cost, Time, and Value of Engineering-Construction Projects*, Wiley, New York.

Hall, E.A., Rosenthal, J. and Wade, J. (1994), 'How to Make Reengineering Really Work', *The McKinsey Quarterly,* Vol. 2, pp. 107-128.

Hammer, M. and Champy, J. (1993), *Reengineering the Corporation: a Manifesto for Business Revolution* , Nicholas Brealey, London .

Hansen, G.A. (1994), *Automating Business Process Reengineering: Breaking the TQM Barrier*, Prentice Hall, New Jersey, USA.

Hayes, R.E. (1998), *Measuring Customer Satisfaction: Survey Design, Use, and Statistical Analysis Methods*, ASQC Quality Press, Milwaukee.

Hicks, D. (1999), *Activity-Based Costing Making It Work For Small and Mid-Sized Companies* , Wiley, New York.

Hofacker, I. and Vetschera, R. (2001), 'Algorithmical Approaches to Business Process Design', *Computers & Operations Research*, Vol. 28, No. 13, pp. 1253-1275.

Horsch, J.C. (1998), ' Where Concepts and Technologies Meet', *Strategic Finance*, Vol. 79, No. 12, pp. 65.

Howard, A., Kochhar, A. and Dilworth, J. (2000), 'Case Studies Based Developments of a Rule-Base for the Specification of Manufacturing Planning and Control Systems', *International Journal of Production Research*, Vol. 38, No. 12, pp. 2591-2606.

Hwang, C.-L. and Masud, A.S.M. (1979), *Multiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*, Springer-Verlag, Berlin.

Hyde, W.F. (1981), *Improving Productivity by Classification, Coding and Database Standardization*, Marcel Dekker, New York.

Ince, M. (2000), *Information Systems: a Methodology for Capturing User Requirements (PhD Thesis)*, Cranfield University, UK.

Innes, J., Mitchell, F. and Yoshikawa, T. (1994), *Activity Costing for Engineers*, Research Studies Press, New York.

Jankowicz, A.D. (1995), *Business Research Projects*, Chapman and Hall, London.

Jiantao, Z., Meilin, S. and Xinming, Y. (2005), 'Formal Verification Techniques in Workflow Process Modeling ', *Journal of Computer Research and Development*, Vol. 42, No. 1, pp. 1-9.

Johansson, H.J., McHugh, A.P., Pendlebury, J. and Wheeler, W.A. (1993), *Business Process Reengineering Breakpoint Strategies for Market Dominance*, Wiley, Chichester.

Johnson, W.C. and Weinstein, A. (2004), *Superior Customer Value in the New Economy : Concepts and Cases,* CRC Press, Boca Raton, Florida, USA.

Kampf, R. and Grobmann, B. (2006), *Workflow Management*, available at: http://www.ebz-beratungszentrum.de/organisation/themen/workflowtext.html (accessed 2008).

Keeney, R.L. and Raiffa, H. (1976), *Decisions With Multiple Objectives: Preferences and Value Tradeoffs,* Wiley, New York.

Kirk, J.L. and Miller, M. (1986), *Reliability and Validity in Qualitative Research*, Sage, California.

Kline, J.J. (2003), 'Activity-Based Costing and Benchmarking: A Tandem for Quality-Oriented Governments', *The Journal of Government Financial Management,* Vol. 52, No. 3, pp. 50-54.

Koubarakis, M. and Plexousakis, D. (2002), 'A Formal Framework for Business Process Modelling and Design', *Information Systems,* Vol. 27, No. 5, pp. 299-319.

Koza, J.R. (1992), *Genetic Programming: on the Programming of Computers by Means of Natural Selection,* MIT Press, Cambridge, MA.

Kusiak, A. (2000), 'Evolutionary Computation and Data Mining ', in B. Gopalakrishnan and A. Gunasekaran *SPIE Conference on Intelligent Systems and Advanced Manufacturing, November 2000,* Vol. 4192, Boston, MA.

Lakin, R., Capon, N. and Botten, N. (1996), 'BPR Enabling Software for the Financial Services Industry', *Management Services*, Vol. 40, No. 3, pp. 18-20.

Library and Archives Canada (LAC) - Information Management Services (2004), *Business Activity Structure Classification System (BASCS)*, available at: http://www.collectionscanada.ca/information-management/0630_e.html (accessed 2006).

Lindsay, A., Downs, D. and Lunn, K. (2003), 'Business Processes - Attempts to Find a Definition', *Information and Software Technology*, Vol. 45, No. 15, pp. 1015-1019.

Man, K.F., Tang, K.S., Kwong, S. and Halang, W.A. (1997), *Genetic Algorithms for Control and Signal Processing*, Springer, London.

May, T. (2002), *Qualitative Research in Action*, Sage, London.

Mayer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.S., Blinn, T., and Perakath, B. (1995), *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*, Knowledge Based Systems Inc., TX.

McCall, J. (2005), 'Genetic Algorithms for Modelling and Optimisation', *Journal of Computational and Applied Mathematics*, Vol. 184, No. 1, pp. 205-222.

Melao, N. and Pidd, M. (2000), 'A Conceptual Framework for Understanding Business Processes and Business Process Modelling', *Information Systems Journal*, Vol. 10, No. 2, pp. 105-129.

Miettinen, K. (1999), *Nonlinear Multiobjective Optimization*, Kluwer, Boston.

Mihelis, G., Grigoroudis, E., Siskos, Y., Politis, Y. and Malandrakis, Y. (2001), 'Customer Satisfaction Measurement in the Private Bank Sector', *European Journal of Operational Research*, Vol. 130, No. 2, pp. 347-360.

NASA (2002a), *Cost Estimating Handbook*, available at: www.jsc.nasa.gov/bu2/NCEH/index.htm (accessed 2005a).

NASA (2002b), *Parametric Estimating Handbook,* available at: http://cost.jsc.nasa.gov/PCEHHTML/pceh.htm (accessed 2008b).

National Archives of Australia (2000), *Business Classification Systems,* available at: http://www.naa.gov.au/recordkeeping/control/tools/appendixB.html (accessed 2006).

Object Management Group (OMG) (2006), *Business Process Modelling Notation (BPMN) Specification: Final Adopted Specification*, OMG, Needham, MA, USA.

Office of Information Technology (OIT) (2005) *Activity Based Cost Estimation,* available at: www.faa.gov (accessed 2005).

Ould, M. (1995), *Business Processes Modelling and Analysis for Re-Engineering and Improvement*, Wiley, Chichester.

Oxford English Dictionary (2006), available at: www.askoxford.com (accessed 2006).

Phalp, K. and Shepperd, M. (2000), 'Quantitative Analysis of Static Models of Processes', *The Journal of Systems and Software,* Vol. 52, No. 2, pp. 105-112.

Picot, A., Reed, B. and Upward, F. (2004), *Workflow - Business Process Re-Engineering,* Monash University, Caulfield, Australia.

Razavian, M. and Ramtin, K. (2008), 'Modeling Variability in Business Process Models Using UML', *Fifth International Conference on Information Technology: New Generations,* Vol. 5, pp 82-87, Las Vegas, USA, Institute of Electrical and Electronics Engineers Inc., New Jersey, USA.

Riordan, R.M. (1999), *Designing Relational Database Systems,* Microsoft Press, Washington, USA.

Robson, C. (2002), *Real World Research*, Blackwell, Malden, MA.

Rossi, P.H. and Freeman, H.E. (1993), *Evaluation: A Systematic Approach*, Sage, USA.

Roy, R. (2003a), *Cost Engineering: Why, What and How?,* Cranfield University, Cranfield, UK.

Roy, R. (2003b), *MSc IT for Product Realisation, Cranfield University - Lecture Materials: Product Evaluation & Manufacture: Cost Engineering,* Cranfield University, UK.

Roztocki, N. (2005), *Activity Based Cost Estimation,* available at: www.pitt.edu (accessed 2005).

Rutenbar, R.A. (1989), 'Simulated Annealing Algorithms: an Overview', *IEEE Circuits and Devices Magazine,* Vol. 5, No. 1, pp. 19-26.

Salimifard, K. and Wright, M. (2000), 'Petri Net-Based Modelling of Workflow Systems: An Overview', *European Journal of Operational Research,* Vol. 134, No. 3, pp. 664-676.

Sayal, M., Casati, F., Dayal, U., and Shan, M.-C. (2002), 'Business Process Cockpit (Hewlett-Packard)', *28th VLDB Conference,* Hong Kong, China.

Scheer, A.-W. (1998), *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer-Verlag, Berlin.

Scheer, A.-W. (2004), *Business Process Automation - ARIS in Practice*, Springer-Verlag, Heidelberg.

Shepperd, M. and Schofield, C. (1997), 'Estimating Software Project Effort Using Analogies', *IEEE Transactions on Software Engineering,* Vol. 23, No. 12, pp. 736-743.

Sivaloganathan, S., Jebb, A., and Evbuomwan, N.F.O. (1998), 'Design for Cost Within Taxonomy of Design Function Deployment', *2nd International Conference on Concurrent Engineering and Electronic Design Automation*, Bournemouth, UK, pp. 14-19.

Smith, L.A. and Mandakovic, T. (1985), 'Estimating: The Input into Good Project Planning', *IEEE Transactions on Engineering Management,* Vol. 32, No. 4, pp. 181-185.

Stelling, M.T., Roy, R., Tiwari, A., and Majeed, B. (2006), 'A Coding Mechanism for Business Process Optimisation Using Genetic Algorithms: Initial Thoughts', *10th IASTED International Conference on Artificial Intelligence and Soft Computing,* Palma, Majorca, ACTA Press, CA.

Stelling, M.T., Roy, R., Tiwari, A. and Majeed, B. (Submitted Sept. 2008a), 'Development of an Automated Business Process Reengineering Approach', *Computers in Industry.*

Stelling, M.T., Roy, R., Tiwari, A. and Majeed, B. (Accepted for publication Dec. 2008b), 'Evaluation of Business Processes Using Probability Driven Activity Based Costing', *The Service Industries Journal.*

Stelling, M.T., Roy, R., Majeed, B. and Tiwari, A. (Provisionally accepted Oct. 2008c), 'Automated Business Process Reengineering: a Methodology for Detecting Risk of Negative Impact on Customer Satisfaction', *BT Innovation Journal.*

Stelling, M.T., Roy, R., and Tiwari, A. (Publication date March 2009), 'A Novel Modelling & Optimisation Technique for Business Processes: An Evolutionary Computing Based Approach', *WSC12 Online Conference,* Springer, London.

Stewart, R., Wyskidsa, R. and Johannes, J. (1995), *Cost Estimator's Reference Manual: 2nd Edition*, Wiley Interscience, Chichester.

Tan, W., Shen, W. and Zhao, J. (2007), 'A Methodology for Dynamic Enterprise Process Performance Evaluation', *Computers in Industry,* Vol. 58, No. 5, pp. 474-485.

Tapscott, D. and Caston, A. (1993), *Paradigm Shift: The Promise of Information Technology,* McGraw-Hill, New York.

Teorey, T.J. (1994), *Database Modeling & Design: The Fundamental Principles*, Morgan Kaufmann, San Francisco.

Towers, S. (1994), *Business Process Re-Engineering : a Practical Handbook for Executives,* Stanley Thomas, Cheltenham, UK.

University of Tennessee (2006), *Biodiversity,* available at: http://web.bio.utk.edu (accessed 2006).

University of Toronto Enterprise Integration Laboratory (2005), *Tools for Business Process Reengineering,* available at: http://www.eil.utoronto.ca/tool/BPR.html (accessed 2008).

Upchurch, A. (2002), *Cost Accounting - Principles and Practice*, Pearson Education, Harlow.

van der Aalst, W.M.P. (1998), 'The Application of Petri Nets to Workflow Management', *Journal of Circuits, Systems and Computers,* Vol. 8, No. 1, pp. 21-66.

van der Aalst, W.M.P., Weijters, A.J.M.M. and Maruster, L. (2004), 'Workflow Mining: Discovering Process Models From Event Logs', *IEEE Transactions on Knowledge and Data Engineering (TKDE),* Vol. 16, No. 9, pp. 1128-1142.

Vergidis, K. (2005), *Multi-Objective Optimisation of Business Processes Using Soft Computing Techniques (MSc Thesis),* Cranfield University, UK.

Vergidis, K., Tiwari, A. and Majeed, B. (2007), 'Business Process Analysis and Optimization: Beyond Reengineering', *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews): Accepted for Future Publication,* No. 99.

Vergidis, K., Tiwari, A., Majeed, B. and Roy, R. (2007), 'Optimisation of Business Process Designs: An Algorithmic Approach With Multiple Objectives ', *International Journal of Production Economics,* Vol. 109, No. 1-2, pp. 105-121.

Wainwright, C. (2004), *MSc Knowledge Management, Cranfield University - Lecture Materials: Process Modelling,* Cranfield University, UK.

Walters, J. (2005), 'Checking of Models Built Using a Graphically Based Formal Modelling Language', *The Journal of Systems and Software*, Vol. 76, No. 1, pp. 55-64.

Wiktionary (2006), available at: http://en.wiktionary.org (accessed 2006).

Workflow Management Coalition (WfMC) (1997), *Workflow Handbook 1997,* John Wiley, New York, USA.

Yin, R.K. (2003), *Case Study Research - Design and Methods,* Sage, London.

Zakarian, A. (2001), ' Analysis of Process Models: A Fuzzy Logic Approach', *International Journal of Advanced Manufacturing Technology,* No. 17, No. 6, pp. 444-452.

Zakarian, A. and Kusiak, A. (2000), 'Analysis of Process Models', *IEEE Transactions on Electronics Packaging Manufacturing,* Vol. 23, No. 2, pp. 137-147.

Zhang, Y.F. (1996), 'Feature-Based Cost Estimation for Packaging Products Using Neural Networks', *Computers in Industry,* Vol. 32, No. 1, pp. 95-113.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix A: Cost Estimating Survey

Industry Survey on Cost Estimation

Cranfield University
School of Applied Sciences

Mark Stelling (m.t.stelling@cranfield.ac.uk)

Tel: 01234-750111 ext. 2858

Interviewee Details

Name: ………………………………………………………………………………
Job Title: ……………………………………………………………………………
Organisation: ……………………………………………………………………….
Address: ……………………………………………………………………………
……………………………………………………………………………………..
……………………………………………………………………………………..
Tel: ....
Email: ………………………………………………………………………………
Years of Experience: ………………………………………………………………

<u>Please note: All information provided will only be used for academic and research purposes</u>

Q1. How do you analyse and predict the operational cost of a business process?

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

Q2. Could you please describe the process? For instance:

- *What are the inputs/outputs to the process?*
- *Do you have examples?*

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

Q3. What are the main challenges in predicting the operational cost?

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q4. Do you use any software based models for the cost prediction?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q5. How do you collect relevant data for the cost prediction, and at what level do you collect it (e.g. task, activity, process, department etc.)?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q6. If you are using ABC, what do you consider to be its positive and negative aspects?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q7. How do you communicate cost estimates to your customers?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q8. If you are using ABC, can you describe the roles of the people involved with it and the information they use?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Q9. What do you consider to be a ‘good’ estimate for operational costs? For instance:

- *How do you recognise one?*
- *What criteria do you use?*

- *What is the expected accuracy level of your estimates?*

…………………………………………………………………………………..

…………………………………………………………………………………..

…………………………………………………………………………………..

…………………………………………………………………………………..

…………………………………………………………………………………..

…………………………………………………………………………………..

Additional Comments:

Thank you very much for your time!

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix B: Customer Satisfaction Questionnaire

Interviewee Details

Name: ………………………………………………………………………………….
Job Title: ……………………………………………………………………………
Organisation: ……………………………………………………………………….
Address: ……………………………………………………………………………….
………………………………………………………………………………………..
………………………………………………………………………………………..
Tel: ....
Email: ………………………………………………………………………………
Years of Experience: ……………………………………………………………….

Please note: All information provided will only be used for academic and research purposes

1. How do you currently measure/monitor customer satisfaction?

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

2. What categories do you use in your measurements (e.g. Image, Service)?

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

3. What aspects of service do you consider most important?

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

4. Which software tools do you use to assist measurement/monitoring?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

5. How would you attempt to measure customer satisfaction levels without customer feedback or customer related data?

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

……………………………………………………………………………..

Appendix C: Validation Questionnaire

Automated Business Process Reengineering Technique: Validation Questionnaire

Technique Components:

Modelling Technique

1) Taxonomy creation leading to unique identifier which allows tracing of every process step to point of origin

	Poor 1				Excellent 5
		2	3	4	
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

2) Process coding system

	Poor 1				Excellent 5
		2	3	4	
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

3) Translation from alternative process model (IDEF3, ARIS) to BPMN, then BPMN to codes

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

Automated Reengineering

4) Stages of the technique (e.g. replicate guidelines of business process reengineering as much as possible without human expert judgement):

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

5) The GA based module:

a. Maximises search space to enable generation of potentially every possible available process

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

b. Generation of feasible processes only

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

6) Example processes – realistic and/or explanatory

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

7) Reengineered example output processes are superior to input

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

Cost Estimating

8) ABC structure (e.g. as per literature/survey)

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

9) Cost drivers/method of allocating costs to direct resources, charged per transaction, and then overheads charged by rate per unit of time

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

10) Varying process resource costs based on probability of usage – to differentiate between processes that use the same resources, but in varying degrees

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

11) Automated method of estimating process cost (e.g. virtual array, stacking of probabilities)

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

12) Method of allocating fitness score (e.g. Delta Cost)

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

Customer Satisfaction

13) ‘Cycle Time’ and ‘Right First Time’ criteria

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

14) Utilisation of all information held in process models to achieve indication of negative or positive effect

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

15) This is potentially the only automated means of providing an indication without feedback from customer or customer related data

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

16) The customer satisfaction formula

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

17) Use of value produced by formula (e.g. slight edge in selection for process with positive; reject negative)

	Poor 1				Excellent 5
		2	3	4	
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

Data

18) Database structure (e.g. reflects taxonomy structure, enables cost estimating framework)

	Poor 1				Excellent 5
		2	3	4	
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

19) Information content

	Poor				Excellent
	1	2	3	4	5
Effectiveness					
Accuracy					
Usability					
Strengths					
Weaknesses					

The Integrated Technique

20) Meets its aim – to develop a framework to automate business process reengineering using cost, duration and impact on customer satisfaction as criteria

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

21) Key objectives achieved / implementation of:

a. Development of an automated reengineering plus optimisation approach for business processes using cost, duration and customer satisfaction criteria

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

b. Development of a mechanism for representing a process model to allow input to an algorithm

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

c. Development of an operational cost modelling approach for business processes and a means of detecting impact on customer satisfaction

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

d. Development of a prototype software platform to validate the approach

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

e. Once implemented, with periodic maintenance can always have optimum available processes in operation

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

f. Implementation issues no worse than any other new system implementation in large companies

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

g. Can be applied to any industry – not only the service sector

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

22) Time and Problem Complexity:

- The example ‘Place Telephone Order’ finds the optimum process in average 20 seconds
- The generic example finds the optimum process in average 85 seconds

a. It is proposed that this technique is significantly faster than a human/manual attempt at optimisation

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

b. Given that the database for a large organisation will contain very large amounts of process data, it is proposed that a human/manual attempt at optimisation may be impossible (e.g generic process example with 129,140,163 possible combinations, each step varying in cost and duration)

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
Comments				

Respondent Profile

Name:
Job Title:

Experience/knowledge of:

- **Process Modelling/Architecture**

Poor **Excellent**

1	2	3	4	5

- **Process Reengineering**

Poor **Excellent**

1	2	3	4	5

- **Conventional & Evolutionary Computing**

Poor **Excellent**

1	2	3	4	5

- **Database Design**

Poor **Excellent**

1	2	3	4	5

- **Data Mining**

Poor **Excellent**

1	2	3	4	5

- **Cost Estimating**

Poor **Excellent**

1	2	3	4	5

- **Measuring/Monitoring Customer Satisfaction**

Poor **Excellent**

1	2	3	4	5

Thank You!

Mark Stelling
m.t.stelling@cranfield.ac.uk

Appendix D: Database Tables

Table: Activity

Columns

Name	*Type*	*Size*
id	Long Integer	4
DomainL3Code	Text	2
Name	Text	50
ActivityCentreCode	Long Integer	4
ActivityCode	Text	2

Table: ActivityCentre

Columns

Name	*Type*	*Size*
id	Long Integer	4
Name	Text	50

Table: Constraints

Columns

Name	*Type*	*Size*
id	Long Integer	4
Code	Text	3
Name	Text	50

Table: DomainL1

Columns

Name	*Type*	*Size*
id	Long Integer	4
Name	Text	50
DomainL1Code	Text	1

Table: DomainL2

Columns

Name	Type	Size
id	Long Integer	4
DomainL1Code	Text	1
Name	Text	50
DomainL2Code	Text	1

Table: DomainL3

Columns

Name	Type	Size
id	Long Integer	4
DomainL2Code	Text	1
Name	Text	50
DomainL3Code	Text	2

Table: Flows

Columns

Name	Type	Size
id	Long Integer	4
Code	Text	2
Name	Text	100

Table: GatewayProbabilities

Columns

Name	Type	Size
CodeID	Text	10
Name	Text	100
OptionProb1	Long Integer	4
OptionProb2	Long Integer	4
OptionProb3	Long Integer	4
OptionProb4	Long Integer	4
OptionProb5	Long Integer	4

Table: Inputs

Columns

Name	***Type***	***Size***
id	Long Integer	4
Code	Text	5
Name	Text	50

Table: Outputs

Columns

Name	***Type***	***Size***
id	Long Integer	4
Code	Text	5
Name	Text	50

Table: OverheadCosts

Columns

Name	***Type***	***Size***
id	Long Integer	4
ServiceDesign	Double	8
InstallationandTesting	Double	8
Hosting	Double	8
SupportandMaintenance	Double	8

Table: ProcessSequences

Columns

Name	***Type***	***Size***
CodeID	Text	10
SeqNo	Long Integer	4
Name	Text	50
DomainL1Code	Text	1
DomainL2Code	Text	1
DomainL3Code	Text	2
ActivityCode	Text	2
TransactionCode	Text	2
SubTransactionCode	Text	2
InputCode(1)	Text	5
InputCode(2)	Text	5
InputCode(3)	Text	5
OutputCode(1)	Text	5
OutputCode(2)	Text	5
OutputCode(3)	Text	5
ConstraintCode(1)	Text	3
ConstraintCode(2)	Text	3
ConstraintCode(3)	Text	3
ResourceCode	Text	3
FlowsIn(1)	Text	2
FlowsIn(2)	Text	2
FlowsIn(3)	Text	2
FlowsIn(4)	Text	2
FlowsIn(5)	Text	2
FlowsOut(1)	Text	2
FlowsOut(2)	Text	2
FlowsOut(3)	Text	2
FlowsOut(4)	Text	2
FlowsOut(5)	Text	2
PreviousStep(1)	Text	10
PreviousStep(2)	Text	10
PreviousStep(3)	Text	10
PreviousStep(4)	Text	10
PreviousStep(5)	Text	10
NextStep(1)	Text	10
NextStep(2)	Text	10
NextStep(3)	Text	10
NextStep(4)	Text	10
NextStep(5)	Text	10
StepType	Text	2
Time	Text	3

Table: ResourceCentre

Columns

Name	***Type***	***Size***
id	Long Integer	4
Name	Text	50

Table: ResourceCosts

Columns

Name	***Type***	***Size***
id	Long Integer	4
ResourceCode	Text	3
ResourceCentreCode	Long Integer	4
ActivityCentreCode	Long Integer	4
ResourceName	Text	50
Cost	Double	8

Table: StepTypes

Columns

Name	***Type***	***Size***
id	Long Integer	4
Code	Text	2
Name	Text	100

Table: SubProcessLibrary

Columns

Name	***Type***	***Size***
CodeID	Text	10
SeqNo	Long Integer	4
Name	Text	50
DomainL1Code	Text	1
DomainL2Code	Text	1
DomainL3Code	Text	2
ActivityCode	Text	2
TransactionCode	Text	2
SubTransactionCode	Text	2
InputCode1	Text	5
InputCode2	Text	5
InputCode3	Text	5
OutputCode1	Text	5
OutputCode2	Text	5
OutputCode3	Text	5
ConstraintCode1	Text	3
ConstraintCode2	Text	3
ConstraintCode3	Text	3
ResourceCode	Text	3
FlowsIn1	Text	2
FlowsIn2	Text	2
FlowsIn3	Text	2
FlowsIn4	Text	2
FlowsIn5	Text	2
FlowsOut1	Text	2
FlowsOut2	Text	2
FlowsOut3	Text	2
FlowsOut4	Text	2
FlowsOut5	Text	2
PreviousStep1	Text	10
PreviousStep2	Text	10
PreviousStep3	Text	10
PreviousStep4	Text	10
PreviousStep5	Text	10
NextStep1	Text	10
NextStep2	Text	10
NextStep3	Text	10
NextStep4	Text	10
NextStep5	Text	10
StepType	Text	2
Time	Text	3

Table: SubTransaction

Columns

Name	***Type***	***Size***
id	Long Integer	4
DomainL1Code	Text	1
DomainL2Code	Text	1
DomainL3Code	Text	2
ActivityCode	Text	2
TransactionCode	Text	2
Name	Text	50
ResourceCode	Long Integer	4
SubTransactionCode	Text	2
InputCode(1)	Text	5
InputCode(2)	Text	5
InputCode(3)	Text	5
OutputCode(1)	Text	5
OutputCode(2)	Text	5
OutputCode(3)	Text	5
ConstraintCode(1)	Text	3
ConstraintCode(2)	Text	3
ConstraintCode(3)	Text	3
ResourceParamCode	Text	3
FlowsIn(1)	Text	2
FlowsIn(2)	Text	2
FlowsIn(3)	Text	2
FlowsIn(4)	Text	2
FlowsIn(5)	Text	2
FlowsOut(1)	Text	2
FlowsOut(2)	Text	2
FlowsOut(3)	Text	2
FlowsOut(4)	Text	2
FlowsOut(5)	Text	2
PreviousStep(1)	Text	10
PreviousStep(2)	Text	10
PreviousStep(3)	Text	10
PreviousStep(4)	Text	10
PreviousStep(5)	Text	10
NextStep(1)	Text	10
NextStep(2)	Text	10
NextStep(3)	Text	10
NextStep(4)	Text	10
NextStep(5)	Text	10
StepType	Text	2
Time	Text	3

Table: Transaction

Columns

Name	***Type***	***Size***
ID	Long Integer	4
ActivityCode	Text	2
Name	Text	50
TransactionCode	Text	2