Nelson Mandela Metropolitan University

for tomorrow

Department of Computer Science and Information Systems

An Investigation into a Natural Language Interface for Contact Centres

January 2009

Gopal Ravi Sankar

Supervisor: Prof. J.Greyling

Co-Supervisor: Dr. D.Vogts

Submitted in partial fulfilment of the requirements for the degree of Magister Scientiae in Computer Science and Information Systems at the Nelson Mandela Metropolitan University

For my family.

Acknowledgements

Firstly I would like to thank God for answering all my prayers and keeping me strong through the many sleepless nights. I would also like to thank my parents for their words of encouragement and support during the period of my research.

My deepest gratitude to my supervisors Prof. Jean Greyling and Dr. Dieter Vogts for their continuous encouragement and support, especially when everything seemed lost. Their constructive criticism, guidance and encouragement were influential in the completion of this dissertation.

I would also like to thank all the staff members and fellow masters students at the Department of Computer Science and Information Systems for their continued support during the course of the research. A special thanks needs to be extended to Mathys C. du Plessis for his insight into my research and providing fresh ideas whenever they were needed.

The Centre of Excellence deserves a mention for the financial support they provided for the completion of this research.

Finally, to Dr. Jacques Pieterson from the Unit for Statistical Consultation (USC) for making the insensible understandable. His passion and guidance for understanding and teaching statistics made the evaluation phase of this research successful and enjoyable.

Table of Contents

Acknowledgements iii
Table of Contents iv
List of Figures viii
List of Tables x
Summary xii
Chapter 1 - Introduction 1
1.1 Background 1
1.2 Relevance of Research 2
1.3 Research Outline 3
1.3.1 Research Question and Objectives 4
1.3.2 Thesis Statement 4
1.3.3 Scope and Constraints 4
1.3.4 Research Design 5
1.4 Dissertation Structure 5
Chapter 2 - Contact Centres 8
2.1. Introduction 8
2.2. Types of Contact Centres 10
2.2.1. Inbound Contact Centres 11
2.2.2. Outbound Contact Centres 12
2.2.3. Help Desk 12
2.2.4. Service Desk 13
2.3 Contact Centre Architecture 15
2.4 Interactive Voice Response 17
2.5 Challenges regarding Contact Centres 20
2.5.1 Customer Satisfaction 21
2.5.2 Cost Reduction 22
2.5.3 Trade-offs between Customer Satisfaction and Cost Reduction 22
2.6 Conclusions 22
Chapter 3 – Rule-Based Expert Systems 24
3.1 Introduction 24

3.2 Rule-Based Expert Systems 27
3.3 Rule-Based Expert System Architecture 28
3.3.1 Inference in Rule-Based Expert Systems 29
3.3.1.1 Forward-Chaining 30
3.3.1.2 Backward Chaining 31
3.4 Advantages and Disadvantages of Rule-Based Expert Systems 33
3.5 Conclusions 34
Chapter 4 - Natural Language Interfaces 35
4.1 Introduction 35
4.2 Natural Language Interface as a User Interface 37
4.2.1 Conversational Speech Interfaces 38
4.2.1.1 Speech Recognition 38
4.2.2.2 Speech Synthesis 39
4.2.2 Text-Based Natural Language Interfaces 41
4.2.3 Shortcomings of Natural Language Interfaces 43
4.2.4 Benefits of Natural Language Interfaces 44
4.3 ALICE 45
4.3.1 Artificial Mark-up Language (AIML) 45
4.3.2 ALICE Pattern Matching Algorithm 47
4.3.3 Shortcomings of ALICE and AIML 50
4.3.4 Benefits of ALICE and AIML 51
4.4 Applications of Natural Language Interfaces 51
4.4.1 Conversation Speech Interface Applications 52
4.4.2 Text Based Conversational Agents 52
4.5 Conclusions 55
Chapter 5 - Design and Implementation 57
5.1 Introduction 57
5.2 Confirmation of Advantages and Disadvantages of NLI and IVR 59
5.2.1 Knowledge Base Compilation 59
5.2.2 Implementation of Prototypes 61
5.2.2.1 ALICE (NLI) 61
5.2.2.2 IVR (RBS) 62
5.2.3 Evaluation of Prototypes 63
5.2.3.1 Evaluation Methods 64

5.2.3.2 Selection of Participants 65
5.2.3.3 Evaluation Instruments and Procedure 66
5.2.3.4 Results of Evaluation 66
5.3 Hybrid Prototype Models 68
5.3.1 Natural Language Interface with Rule-Based Expert System Concepts 68
5.3.1.1 A Model for a Natural Language Interface with Rule-Based Concepts 72
5.3.2 Rule-Based Expert System using a Natural Language Interface........ 73
5.3.2.1 Model for a Rule-Based Expert System using a Natural Language Interface........ 75
5.4 Model Comparison........ 78
5.5 Implementation of the Hybrid Prototype........ 79
5.5.1 Establishing Structure for AIML........ 79
5.5.2 Implementing AIML and Maintaining Context........ 80
5.5.3 Implementation Issues 82
5.5.4 Pilot Study 82
5.5.4.1 Goals 83
5.5.4.2 Selection of participants 83
5.5.4.3 Development of Tasks 83
5.5.4.4 The test procedure 83
5.5.4.5 Results........ 84
5.6 Conclusions........ 86
Chapter 6 - Evaluation 87
6.1 Introduction........ 87
6.2 Statistical Instruments........ 88
6.2.1 *T*-tests 88
6.2.2 Wilcoxon Matched Pairs test 88
6.2.3 Analysis of Variance (ANOVA)........ 89
6.2.4 Tukey Honestly Significant Difference (HSD) Test........ 89
6.2.5 Spearman Rank Order Correlation 89
6.3 User Testing........ 89
6.3.1 Usability Measurements........ 90
6.3.2 Evaluation Instruments and Procedure........ 91
6.3.3 Selection of Participants 92
6.4 Results of User Testing........ 95
6.4.1 Quantitative Results 95

6.4.2 Inferential Statistics 99
6.4.2.1 Time Taken per Task 100
6.4.2.2 Usability Metrics 102
6.4.2.4 Time Taken versus Usability Metrics 106
6.4.3 Qualitative Analysis 108
6.4.3.1 Negative Comments 108
6.4.3.2 Positive comments 109
6.5 Summary of Results 110
Chapter 7 - Conclusion 112
7.1. Introduction 112
7.2. Research Contributions 113
7.2.1 Theoretical Contributions 116
7.2.2 Practical Contributions 117
7.3 Implications of Research 119
7.4 Future Research 119
7.4.1 Implementation of RBS using NLI Concepts Model 119
7.4.2 Speech Recognition and Synthesis 120
7.4.3 Embodied Conversational Agents 120
7.4.4 Other Techniques to Maintain Context of Conversation 120
7.4.5 Implementation of the Models on a Mobile Phone 120
7.4.6 Automating the Generation of AIML with Rules 121
7.5 Summary 121
References 122
Appendix A 127
Appendix B 131
Appendix C 135
Appendix D 141
Appendix E 142
Appendix F 151

List of Figures

Figure 1.1 Dissertation Structure 6
Figure 2.1 Contact Centre Environment (Gans *et al.* 2003) 9
Figure 2.2 Contact Centre Types and Classification (Singh 2007) 11
Figure 2.3 Contact Centre Architecture (Gans *et al*. 2003) 16
Figure 2.4 Basic Flow of Touch Tone IVR 18
Figure 3.1 Some Areas of Artificial Intelligence (Giarrantano *et al*. 1989) 25
Figure 3.2 Architecture of a Simple RBS (Abraham 2005) 29
Figure 3.3 Representation of a Printer Troubleshooting RBS Rules (Hayes-Roth 1985) 30
Figure 4.1 How TTS Works (Acapela Group 2008) 40
Figure 4.2 A Virtual World Personal Assistant 42
Figure 4.3 AIML Format with Wildcards 46
Figure 4.4 AIML Predicates 46
Figure 4.5 Predicate Values 46
Figure 4.6 The <SRAI> Tag 47
Figure 4.7 Branches of ALICE Brain with 24000 Categories Loaded. 49
Figure 4.8 The Hanover Chatbot 53
Figure 4.9 Best Western Chatbot 54
Figure 4.10 Virgin Holidays Chatbot 55
Figure 4.11 Jeeney AI 55
Figure 5.1 Breakdown of Data Gathering Sample in Accordance to Occupation (n=15) 60
Figure 5.2 Gender Profile for Data Gathering Sample (n=15) 61
Figure 5.3 Chat History with a Data Gathering Participant 61
Figure 5.4 ALICE System utilising an IM Interface 62
Figure 5.5 A Rule in CLIPS 63
Figure 5.6 RBS Utilising an IVR Approach 63
Figure 5.7 Breakdown of Evaluation Sample in Accordance to Occupation (n=16) 65
Figure 5.8 Gender Profile for Evaluation Sample (n =16) 66
Figure 5.9 Simplified Graph Structure of ALICE 69
Figure 5.10 Conflicts with the Same Pattern in AIML 70
Figure 5.11 Simplified Graph Structure with Rules Compared to IVR 71
Figure 5.12 AIML with Rules 72
Figure 5.13 Rule Based Natural Language Interface Model (Ravi Sankar *et al.* 2008b) 73
Figure 5.14 Graph Structure of an IVR 74
Figure 5.15 Inner Working of the Rule Based Expert System using a Natural Language Interface 75
Figure 5.16 Graph Structure of an NLI for RBS 76
Figure 5.17 Rule Based Expert System using a Natural Language Interface Model (Ravi Sankar *et al.* 2008b) 77
Figure 5.18 Menu Structure Utilised to Create Rules 81
Figure 6.1 Gender Profile of Test Participants (n=20) 92
Figure 6.2 Age Profile of Test Participants (n=20) 93
Figure 6.3 Computing Experience Profile of Participants (n=20) 93

Figure 6.4 Printer Experience Profiles of Test Participants (n=20).. 94
Figure 6.5 Printer Troubleshooting Experience of Test Participants (n=20)...................................... 94
Figure 6.6 Instant Messaging Experience Profiles of Test Participants (n=20) 95
Figure 6.7 IVR Participant Quantitative Responses (n=20) 96
Figure 6.8 NLI Participant Quantitative Responses (n=20) 97
Figure 6.9 Hybrid Participant Quantitative Responses (n=20)... 98
Figure 6.10 Ranking of Prototypes 99
Figure 6.11 Time on Task across the Three Prototypes.. 100
Figure 6.12 Usability of the Three Prototypes 103
Figure 7.1 NLI using RBS concepts Model... 117
Figure 7.2 Menu Structure Utilised to Create Rules.. 118
Figure 7.3 Custom Tags as used in the Hybrid Prototype... 119

List of Tables

Table 2.1 Advantages and Disadvantages of Centralised and Decentralised Help Desks and Service Desks 14
Table 3.1 Characteristics of Forward and Backward Chaining 33
Table 3.2 Summary of Advantages and Disadvantages of RBS 34
Table 5.1 Advantages and Disadvantages of NLI and RBS 58
Table 5.2 Questions Used to Evaluate Usability Metrics 65
Table 5.3 Top 3 Problems Encountered when Utilising NLI (Ravi Sankar, Greyling and Vogts 2008a) 67
Table 5.4 Top 3 Problems Encountered when Utilising IVR (Ravi Sankar *et al.* 2008a) 67
Table 5.5 Comparison of NLI and IVR (Ravi Sankar *et al.* 2008a) 67
Table 5.6 Comparison of Hybrid Models 78
Table 6.1 Statements used in Post-Test Questionnaire and their Relation to Usability 90
Table 6.2 Questions asked in the Post-Test Questionnaire for the Three Prototypes 96
Table 6.3 T-Test and Wilcoxon Matched Pairs Test p Values for Task1 101
Table 6.4 T-Test and Wilcoxon Matched Pairs Test p Values for Task2 101
Table 6.5 T-Test and Wilcoxon Matched Pairs Test p Values for Task3 102
Table 6.6 Repeated Measure ANOVA Test p Values for Prototypes 103
Table 6.7 Tukey HSD Test for ALICE Prototype 104
Table 6.8 Repeated Measure ANOVA Test p Values for Prototypes 104
Table 6.9 Tukey HSD Test Showing Significant Difference between the Prototypes with the effectiveness metric 105
Table 6.10 Tukey HSD Test showing Significant Difference between the Prototypes with the Efficiency Metric 106
Table 6.11 Tukey HSD Test showing Significant Difference between the Prototypes with the Satisfaction Metric 106
Table 6.12 Spearman Rank Order Correlations for Usability Metrics on IVR 107
Table 6.13 Spearman Rank Order Correlations for Usability Metrics on IVR 107
Table 6.14 Spearman Rank Order Correlations for Usability Metrics on Hybrid 107
Table 6.15 Summary of advantages and disadvantages of three prototypes 110
Table 7.1 Summary of Limitations and Benefits Addressed by the Three Prototypes 115
Table 7.2 Summary of Advantages and Disadvantages of NLI and IVR 116
Table D.1 Biographical Data of Test Participants (N=20) 141
Table E.1 IVR User Likert Scale Ratings 142
Table E.2 ALICE User Likert Scale Ratings 142
Table E.3 Hybrid User Likert Scale Ratings 142
Table E.4 IVR User Ranking 143
Table E.5 ALICE User Ranking 143
Table E.6 Hybrid User Ranking 143
Table E.7 Comparison of Three Prototype Rankings 143
Table E.8 T-test comparing Time Taken to complete Task 1 on IVR and Hybrid Prototypes 143
Table E.9 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 1 on IVR and Hybrid 144

Table E.10 T-test comparing Time Taken to complete Task 1 on Hybrid and ALICE Prototypes 144
Table E.11 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 1 on Hybird and ALICE 144
Table E.12 T-test comparing Time Taken to complete Task 1 on IVR and ALICE Prototypes 144
Table E.13 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 1 on IVR and ALICE 144
Table E.14 T-test comparing Time Taken to complete Task 2 on IVR and Hybrid Prototypes 145
Table E.15 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 2 on IVR and Hybrid 145
Table E.16 T-test comparing Time Taken to complete Task 2 on Hybrid and ALICE Prototypes 145
Table E.17 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 2 on Hybrid and ALICE 145
Table E.18 T-test comparing Time Taken to complete Task 2 on IVR and ALICE Prototypes 145
Table E.19 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 2 on IVR and ALICE 146
Table E.20 T-test comparing Time Taken to complete Task 3 on IVR and Hybrid Prototypes 146
Table E.21 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 3 on IVR and Hybrid 146
Table E.22 T-test comparing Time Taken to complete Task 3 on Hybrid and ALICE Prototypes 146
Table E.23 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 3 on Hybird and ALICE 146
Table E.24 T-test comparing Time Taken to complete Task3 on IVR and ALICE Prototypes 147
Table E.25 Wilcoxon Matched Pairs Test comparing Time Taken to complete Task 3 on IVR and ALICE 147
Table E.26 Comparison of Usability Metrics across Three Prototypes 147
Table E.27 Time Taken VS Likert Scale Rating for IVR 147
Table E.28 Time Taken VS Likert Scale Rating for NLI 148
Table E.29 Time Taken VS Likert Scale Rating for Hybrid 148
Table E.30 Anova Test for IVR 148
Table E.31 Anova Test for NLI 148
Table E.32 Tukey HSD Test for NLI 149
Table E.33 Anova Test for Hybrid 149
Table E.34 Anova Test for Effectiveness 149
Table E.35 Tukey HSD Test for Effectiveness 149
Table E.36 Anova Test for Efficiency 150
Table E.37 Tukey HSD Test for Efficiency 150
Table E.38 Anova Test for Satisfaction 150
Table E.39 Tukey HSD Test for Satisfaction 150

Summary

Contact centres are the first point of contact between a company and a customer after the purchase of a product or service. These centres make use of contact centre agents to service customer queries. In the past contact centres hired as many agents as they could in order to service customers, which have led to an increase in personnel costs causing contact centres to become costly to run.

Automation techniques were introduced to decrease personnel costs and one such technique is the Interactive Voice Response (IVR). The usability of IVR systems is, however, dismal. Customers would rather speak to a contact centre agent than navigate through the menu structure found in these systems. The menu structure has come under scrutiny because it is difficult to use and navigate, is often not aligned to caller usage patterns, and the menu options are long and vague.

This research investigated whether a Natural Language Interface (NLI) could alleviate the problems inherent to IVR. NLIs, however, come with their own disadvantages of which the main ones are ambiguity and the loss of context of a conversation. Two prototypes were implemented, one of which resembled an IVR and the other an NLI (using ALICE concepts). An evaluation of two prototypes confirmed the advantages and disadvantages of these concepts in accordance to theory. A Hybrid prototype was proposed with the aid of two models. The model which proposed an NLI using a rule base was chosen for implementation.

The Hybrid prototype was then evaluated against the NLI and IVR prototypes to deduce which prototype was the most effective, efficient and satisfying. The evaluation through the aid of descriptive and inferential statistics showed that the Hybrid prototype was the most usable prototype.

The evaluation of the Hybrid prototype confirmed that a Hybrid approach could limit the shortcomings of IVR through the elimination of the menu structure found in these systems, thereby allowing users to state their queries in natural language. The incorporated rule base provided the Hybrid system with long term memory, eliminating one of the main disadvantages of NLIs.

Chapter 1 - Introduction

In every phenomenon the beginning remains always the most notable moment -- Thomas Carlyle

1.1 Background

A call centre is an integral part of any business as it provides value added services via the telephone. A contact centre is an extension of a call centre and can be accessed by customers using telephones, email, snail mail, instant messaging and various other communication media (Gans, Koole and Mandelbaum 2003). The variety in communication channels have made contact centres the preferred and prevalent method for customers to receive information from companies. Contact centres are not restricted to a particular business domain and can be implemented by any corporation which is willing to provide additional support and marketing services to their customers (Singh 2007).

A contact centre is therefore, the first point of a customer's contact with a company after the purchase of a product or service. Customer service is highlighted as a critical success factor at these centres. Customers interact with a contact centre through contact centre agents or various automation techniques that are employed there. The most prevalent of these automation techniques is known as Interactive Voice Response (IVR). IVR systems typically utilise touch-tone input from users as a form of input to the various interactive menu options available to customers. IVR provides a self-service option for customers through which they can solve easy or repetitive queries which would otherwise be handled by an expensive contact centre agent (Voxeo 2007).

Rule-Based Expert Systems (RBSs) are specialised systems that use knowledge of a human expert to solve problems (Giarrantano and Riley 1989). Conventional computer programs use problem-solving methods that make use of algorithms, data structures and reasoning to find solutions (Abraham 2005). For more difficult problems, that require the need to simulate human expertise, RBSs constitute a good way to codify the know-how of a human expert (Durkin 1994). Knowledge for these systems is gathered from experts or from general knowledge found in books and magazines. RBSs have been utilised in a variety of domains such as oil exploration, aircraft navigation systems and contact centres to solve problems.

IVR is an example of an RBS, as it utilises human knowledge from contact centre agents and various other contact centre personnel, in order to solve problems at the level of a human expert. This knowledge is embedded in the menus customers encounter when interacting with these systems. IVR has come under scrutiny as they have led to poor interactions with customers and low resolution rates (Saluja 2006). The main reason for these problems was caused by the menu structure utilised by these systems.

Natural language is one of many interface styles that can be used in the dialogue between a human user and a computer through the use of speech, text or gestures (Long 1994). NLIs provide a means through which the menu structure utilised by IVR can be flattened or even eliminated. Natural Language Interfaces (NLIs) are only capable of understanding a restricted subset of human language, which is usually restricted to a certain domain, and generate more or less pre-packaged responses (Patridge 1991).

Conversational agents are a type of NLI that are used to interact with humans and are generally found on retail websites. Customers utilise theses agents to enquire about products or services. This research concentrates on a conversational agent known as ALICE.

1.2 Relevance of Research

IVR is seen as the most prevalent technology in a contact centre (Purushothaman 2004) and is used as a self service option by customers to access data that will assist them to troubleshoot their problems or answer their queries. Though IVR systems seem advantageous to contact centres because of a potential reduction in personnel costs, they have come under scrutiny for two main reasons namely, poor quality of interaction with customers and low resolution of queries (Saluja 2006). The main reasons that IVR systems have such bad reputations are related to their menus which are difficult to navigate and often not aligned to caller usage patterns. Furthermore, unnecessary and irrelevant information is directed at users.

Dimension Data reports that 66% of contact centres utilise IVR to respond to calls outside their operating hours. The abandonment rate once the caller is within the IVR is 12% and the average time of abandonment time is 65 seconds (Dimension Data 2006). Customers therefore dislike IVR and seek contact centre agent assistance at the first opportunity.

NLIs, in particular conversational agents, have the potential to address problems associated with IVR. Since these interfaces allow for natural everyday language to be utilised in the interaction with these systems, the menu structure found in IVR can be flattened or eliminated and users will not feel restricted by the system. Customer interactions with these systems can be effective, efficient and satisfying. Contact centres could save money through reduced contact centre agent personnel costs and greater call densities and, at the same time, build better customer relationships through better interactions.

NLIs have their own share of disadvantages with the main disadvantage being that of ambiguity. A computer can interpret a sentence in a number of different ways in comparison to a human who would take into account the nature and environment of a conversation. Ambiguity can be combated through the utilisation of a variety of techniques of which the most popular used is engaging in a clarification dialogue to confirm if the computer interpretation is the correct one.

From this discussion it becomes evident that an investigation into how NLIs can reduce the limitations of IVR has definite relevance to the automation at contact centres.

1.3 Research Outline

The research outline is separated into four separate sections: research question and objectives, thesis statement, scope and constraints, and research design. These are discussed in more detail in the subsequent subsections.

1.3.1 Research Question and Objectives

The primary research question is:

Can a natural language interface address the limitations and enhance the benefits brought about by interactive voice response as an automation technique at contact centres?

In order to answer this question the following research objectives have been identified:

- Identification of challenges to contact centres and limitations of their automation techniques, with special attention paid to IVR. (Chapter 2)
- Identification of benefits and shortcomings of rule-based expert systems (Chapter 3).
- Identification of the different types of NLIs, their benefits and shortcomings (Chapter 4).
- Implementation and evaluation of two prototypes modelled to resemble an IVR and an NLI to confirm advantages and disadvantages according to theory (Chapter 5).
- Proposal of models for a Hybrid prototype that would combine NLI and IVR concepts and implementation of one of these models (Chapter 5).
- Evaluation of the three prototypes (NLI, IVR and the Hybrid) to determine which one is the most effective, efficient and satisfying (Chapter 6).

1.3.2 Thesis Statement

A Hybrid system that utilises a natural language interface with rule-based expert system concepts will provide a more effective, efficient and satisfying automation approach than current techniques employed at contact centres.

1.3.3 Scope and Constraints

The scope of this research is restricted to text based conversations. Therefore the recognition and synthesis of speech is not investigated in depth. The prototypes that are developed are limited to supporting the diagnosing and resolving of customer queries in the troubleshooting of printer problems. It can be argued that a broader domain is not necessary in answering the main research question. This limitation also decreased the need for an extensive knowledge base to support the developed NLI.

1.3.4 Research Design

This research will focus on the implementation and evaluation of three prototypes that use the concepts of IVR and NLI. The three prototypes are the IVR, NLI and Hybrid. The research methods used in this research will include: literature survey, prototype construction and evaluation.

A literature survey will be used to ascertain the advantages and disadvantages of IVR and NLI. The prototype construction phase will construct two prototypes: NLI and IVR. An initial evaluation will then be conducted to confirm the advantages and disadvantages in accordance to theory. Once this is completed a third prototype will be proposed through two models. One model will be implemented to yield the Hybrid prototype. All three prototypes will then be evaluated through the use of evaluation instruments such as case studies and post-test questionnaires in order to deduce which prototype provides the most effective, efficient and satisfying solution to automation at contact centres. A variety of statistical methods will be utilised to analyse the post-test questionnaires to statistically conclude which is a better solution for contact centres.

1.4 Dissertation Structure

This dissertation will consist of seven chapters. Each chapter will attempt to meet the research objectives mentioned in Section 1.3.1. The structure of the dissertation is shown in Figure 1.1.

Chapter 1 (Introduction) presented a short discussion on contact centres and on the field of NLIs. The preliminary literature study revealed the concerns and problems with IVR as an automation technique at contact centres and discussed how NLIs can be utilised to combat some of the problems brought upon by IVR (Sections 1.1 and 1.2). The main research question and the various objectives were discussed in Section 1.3.1. Scope and constraints of this research were highlighted in Section 1.3.3 and the research design was discussed in the subsequent section (Section 1.3.4).

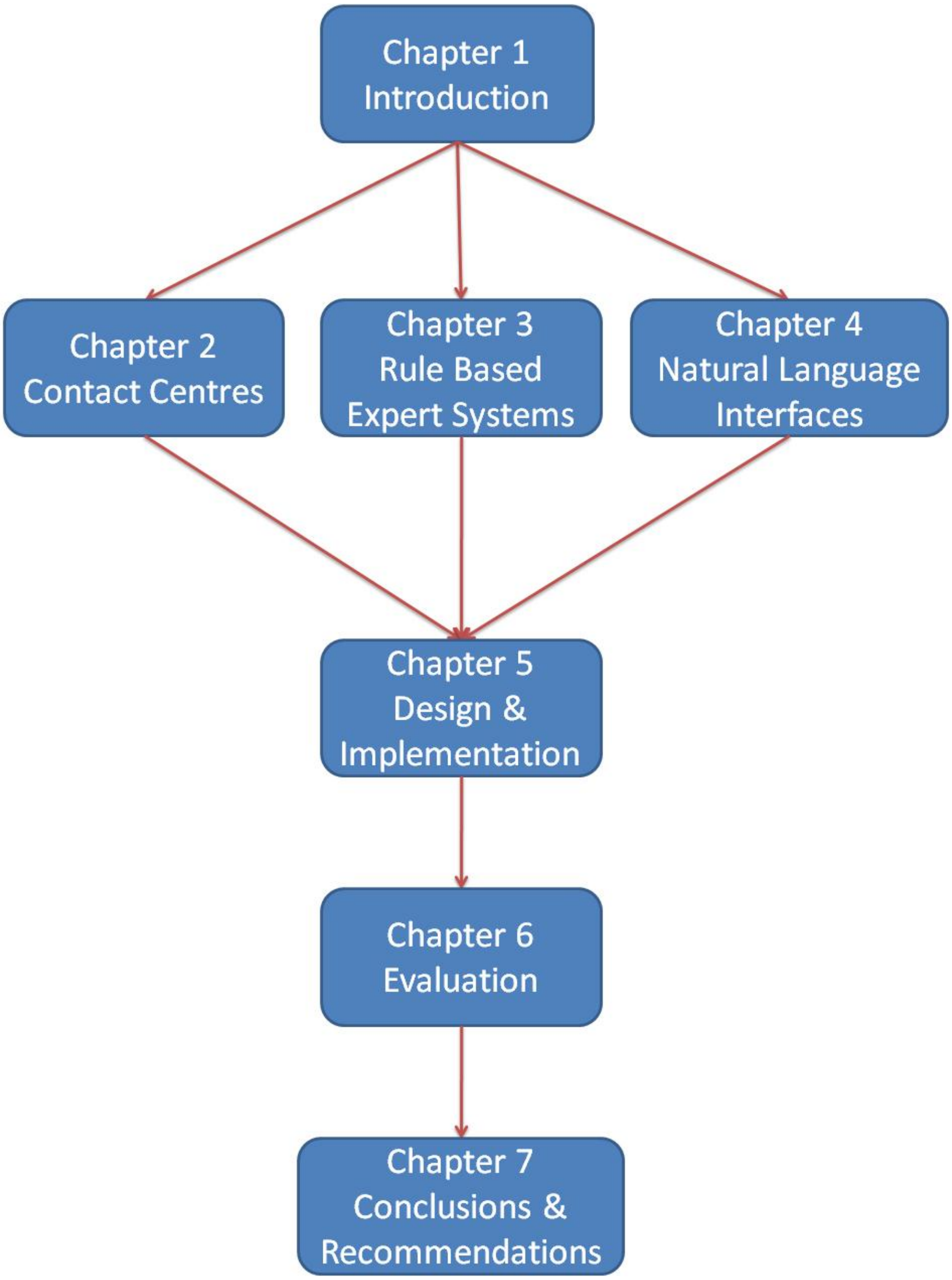

Figure 1.1 Dissertation Structure

Chapter 2 (Contact Centres) is the first literature study chapter presented. This chapter focuses on presenting the different types of contact centres and their architectures. IVR is introduced as one of the components found in the architecture and is discussed in detail, including an overview on the advantages and disadvantages of these systems. Lastly the challenges that contact centres face on a daily basis are discussed.

Chapter 3 (Rule-Based Expert Systems) presents a literature study on rule-based expert systems. It aims to identify the advantages and disadvantages of these expert systems. A discussion on the generic architecture of an expert system is conducted upon which the various inferences that these systems utilise are presented.

Chapter 4 (Natural Language Interfaces) presents a detailed literature study in the field of NLIs. It aims to define what NLIs are and discusses the various advantages and disadvantages that they can provide as a user interface. An overview on the different types of NLIs is also presented with special emphasis given to a text based NLI known as ALICE. Existing NLIs are also presented in order to understand to what extent they are utilised to support contact centre operations and to determine if they are successful.

Chapter 5 (Design and Implementation) focuses on the confirmation of advantages and disadvantages in accordance to literature. The design and implementation of an NLI and IVR and the evaluation of these prototypes are conducted to confirm advantages and disadvantages in accordance to theory. The chapter will also focus on two Hybrid models that utilises NLI and IVR concepts that could be utilised as solution for automation at contact centres. The choice and implementation of one model is presented.

Chapter 6 (Evaluation) discusses the evaluation of the three prototypes in order to determine which of the prototypes (NLI, IVR, Hybrid) is the most effective, efficient and satisfying. The evaluation strategy is presented upon which the results are discussed.

Chapter 7 (Conclusions and Recommendations) presents the conclusions of the dissertation. It examines whether the research objectives set out in Chapter 1 were met. The various achievements and contributions made are highlighted upon which suggestions for future research are discussed. This chapter is followed by a list of references and appendices are provided at the end of the document.

.

Chapter 2 - Contact Centres

The most basic and powerful way to connect to another person is to listen. Just listen. Perhaps the most important thing we ever give each other is our attention... -- Rachel Naomi Remen.

2.1. Introduction

Contact centres are seen as a response to the demand for more convenience in a world that moves faster (Bergevin and Wyatt 2008). Contact centres are an integral part of any business as they are not restricted to a particular business domain and can be implemented by any business willing to provide customer service to their customer (Singh 2007). The difference between call centres and contact centres is the difference in the communication channels used to gain access to these centres. A call centre as its name implies can only be reached via a telephone while a contact centre can be accessed via a variety of communication channels such as the telephone, email, snail mail, instant messaging and various other communication media utilised by customers (Gans *et al.* 2003). The varieties in communication channels have made contact centres the preferred and prevalent means for customers to receive information from companies.

The move from call centres to contact centres has been motivated by societal hype surrounding the Internet and by customer demand for channel variety. This variety has also provided the customer and the contact centre with the potential for efficiency gains (Dawson 2003). The variety of communication channels have benefits to the customer such as requests being stored for later response. This could be cheaper than solving a query on the spot via telephone services (Bergevin *et al.* 2008). Queries and transactions can also be completed in a more efficient as information provided by customers can be stored in a structured manner (Australian National Audit Office 1996).

Contact centres consist of various resources such as personnel, computers, various telecommunications equipment and software which enable the delivery of service to customers. The work environment of a typical contact centre can be envisioned as a room with endless open space cubicles in which contact centre agents with earphones sit in front of computers and provide relevant services to customers (Figure 2.1). A contact centre agent's responsibility is to handle calls placed or received.

Figure 2.1 Contact Centre Environment (Gans *et al.* 2003)

Contact centres are prevalent in two areas (Friedman 2001):

- Customer service and retention – since contact centres are the first point of contact with a company after the purchase of a product or service, they create long term relationships with their respective customers and therefore maintain customer satisfaction.

- Customer relations and marketing data – support provided by a contact centre allows it to acquire information about the customer which could be used for conducting market research, design of a new product or service and contacting customers about value added product offerings.

Despite the benefits of contact centres, they have been dismissed as being very cost intensive due to their high operational and personnel costs (Almskog & Frydman Communications 1996). Contact centres have as a consequence had to reduce the number of personnel they employ which has decreased costs but also reduced customer service. For companies this is a problem because excellent customer service can be seen as a means to attract new customers and retain old ones (Woodward 2007).

This chapter gives an overview on the different types of contact centres (Section 2.2) and the various technologies utilised by them (Section 2.3). One of these technologies is Interactive Voice Response (IVR). Section 2.4 discusses the benefits and shortcomings of this technology. This is followed by a discussion on the challenges that contact centres face on a daily basis (Section 2.5).

2.2. Types of Contact Centres

Contact centres can be classified in a variety of ways and the functions they provide vary from one institution to another. These functions include customer service, help desk, emergency response services, telemarketing and order taking. Contact centres also vary in size and geographic dispersion. For example a small medical practice would make use of a few contact centre agents to take calls, while a large national or international centre could have thousands of agents stationed at various physical locations.

The most popular classifications of contact centres are in accordance to their area of operation and their functionality. There are two main ***types*** of contact centres in accordance to ***functionality*** namely *inbound* (Section 2.2.1) and *outbound* (Section 2.2.2) (Call Centre India 2007). According to Singh (2007), the above classification can be further broken down in accordance to ***areas of operation*** as being a *help desk* (Section 2.2.3) or a *service desk* (Section 2.2.4). Figure 2.2 illustrates the classification of a contact centre by type and areas of operation.

2.2.1. Inbound Contact Centres

Inbound contact centres handle incoming calls from customers calling into a contact centre (Gans et al. 2003; Robertson 2005). These centres provide value added service in the form of customer support, reservations and sales support for products and services sold or leased to customers. Inbound contact centres can be further classified as those that *handle queries* or *transactions* or *both* (Robertson 2005).

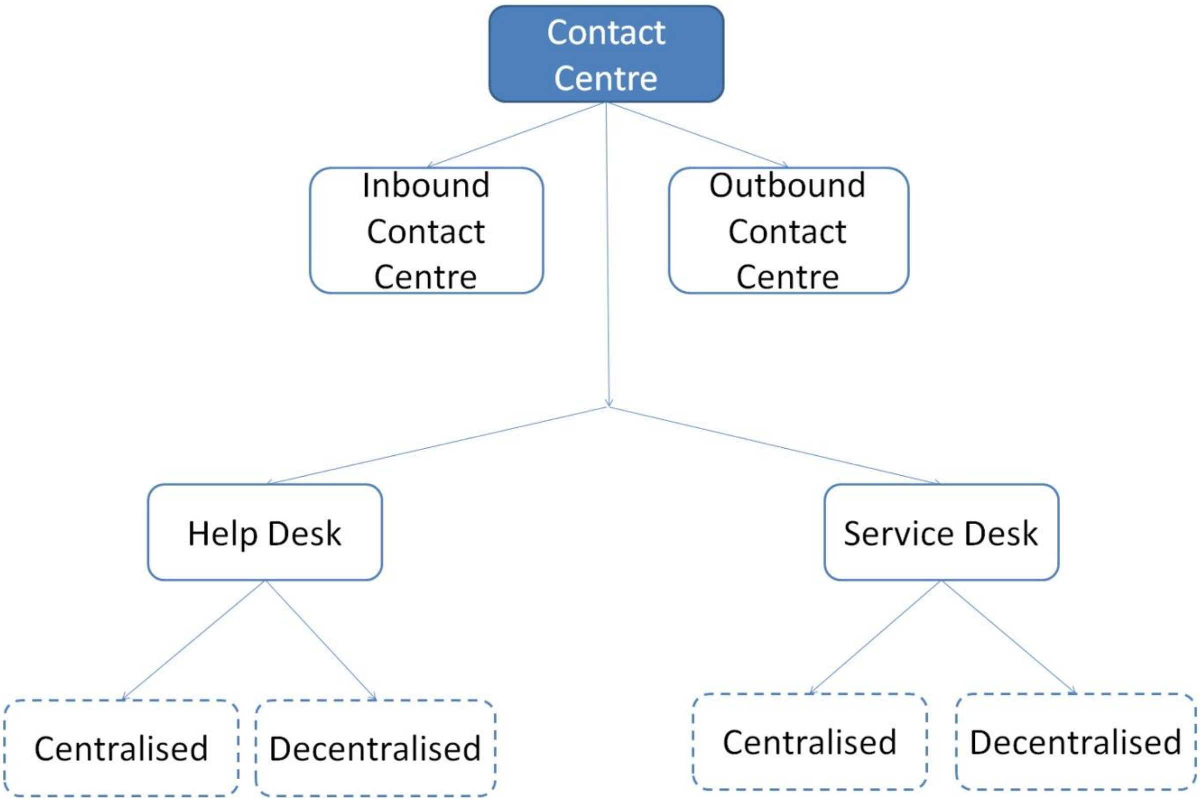

Figure 2.2 Contact Centre Types and Classification (Singh 2007)

Inbound contact centres that *handle queries* are responsible for answering of queries from customers or members of the general public. The queries handled by these centres cover a wide spectrum of subjects. The contact centre agents at these centres would need to have access to a vast knowledge base which they could use to assist them in answering these queries.

Transactional inbound contact centres primarily deal with processing transactions for customers that traditionally may have occurred over a counter. Contact centre agents at transactional inbound contact centres would need initial training as they would need to enter customer data on multiple systems in order to complete a transaction. The agents are trained with a set of procedures to be followed in order to complete a certain transaction (Robertson 2005).

These two types of contact centres do not need to exist in isolation as *both* these inbound contact centres (query and transaction) can be integrated through the use of complex hardware, software and a vast knowledge base (Robertson 2005).

2.2.2. Outbound Contact Centres

Outbound contact centres are responsible for initiating calls from within the centre. These centres are usually found in telemarketing and survey organisations. The main aims of these centres are to sell or advertise a product or service, contact list updating, surveys and verification services (Robertson 2005). Outbound contact centres place calls to highly valued customers who abandoned calls to the centre before being served by a contact centre agent.

2.2.3. Help Desk

A help desk is defined as a single point of contact which deals with the resolution of customer queries and requests (Middleton and Marcella 1996; nanoDesk technologies 2002; Singh 2007). The primary responsibility of a help desk is being the first point of contact that responds immediately to queries posted by customers – internally or externally. The support provided by a help desk deals with customers experiencing problems with day to day tasks or with their information technology system and require support to solve this problem (Contact Centre University 2000).

The first consideration when setting up a helpdesk is whether one location is necessary or if multiple locations are more feasible for the organization. Help desks are therefore either *centralised* or *decentralised* (Sanderson 2003).

A *centralised help desk* is a single physical location with an organisation that provides support to all users and is usually located within the information technology department. The information coming into the help desk goes to the same location and within a short time period, support personnel are exposed to a wide range of problems as they are the only source of technical support within an organisation. The main advantage of this is that support staff within a centralised help desk build a vast knowledge base. The major disadvantages of a centralised help desk are that if the company is located in different time zones, they may require support during different times of the day and not all staff members may understand the unique needs of each business unit (Sanderson 2003).

A *decentralised help desk* is one which consists of multiple sites located throughout an organisation. This type of helpdesk is utilised by organisations that have offices in multiple locations. These offices may be located at various time zones, therefore creating a need for support at times when a centralised help desk may be closed. A decentralised help desk may also be used for specialization. Specialised help desks deal with issues in a specific area of IT such as networking, hardware or software. The advantage of a decentralised helpdesk is that a certain helpdesk may cater for a specific business unit within a company and would be able to build a knowledge base on this specific business unit. A challenge that these help desks face is to provide standardised information to all users (Sanderson 2003). If the help desk does not provide standardised solutions to employees in different divisions, then the organisation may not be utilising their technologies as they would desire. Table 2.1 contains a summary of the advantages and disadvantages of centralised and decentralised help desks.

2.2.4. Service Desk

Service Desks are defined as “a single point of contact for end users who need anything from information technology” (Jones 2005). The Information Technology Infrastructure Library (ITIL) takes the definition further by stating that a service desk extends the range of services provided by a help desk. This allows business processes to be integrated into the service management infrastructure (ITIL Resources 2008).

The service desk has great strategic importance to an organisation as it represents the interests of the customer to the rest of the information technology organisation (Jones 2005). Customer service is therefore seen as an important function and is used as a measurement of effective information technology services management. The service desk can be viewed as a focal point for interaction between customers and the information technology of an organisation. It acts as an interface between the customers and the IT functions (Microsoft 2008).

The main objectives of a service desk are to (Frantz 2006):

- act as the single point of contact for all its information technology customers;

- make sure that service is always available;
- restore service when not available;
- provide support for business critical systems; and
- detect incidents as soon as possible.

	Centralised	Decentralised
Advantages	Easily located by users Communication by specialists easier as they are all located in one area Standards are easily enforced Resource utilisation is better Specialists are readily available for a broad range of issues	Can easily address local site needs (time zone, language, products) Services are available on-site
Disadvantages	Difficult to provide on-site support for remote locations Difficult to understand the business needs of all departments within the organisation Support for departments located in various time zones is difficult	Standardised information is difficult to provide Mission and goals will be different from help desk to help desk Measuring performance will be difficult

Table 2.1 Advantages and Disadvantages of Centralised and Decentralised Help Desks and Service Desks

There are three types of service desks, namely *centralised*, *decentralised* and a *virtual* structure. The *centralised service desk* like its help desk counterpart supports all users within an organisation, regardless of their geographical location (Microsoft 2008). The advantages of a centralised service desk are that users know where to call when they need support, fewer staff may be required (which means less training, equipment and facility costs) and a consolidated management overview (Microsoft 2008). One major disadvantage of this type of service desk is that it may not understand the business needs of the different business units within the organisation.

A *decentralised service desk* has a number of service desks located in various geographic locations. The advantages of this type of service desk are that it provides customised support for the different business units within an organisation and that the support staff can have a broader pool of knowledge to draw information from to solve unique problems (Microsoft 2008) The main disadvantage of this type of service desk is synchronisation of data available at the different service desks.

A *virtual service desk* is a combination of a centralised and a decentralised service desk (ITIL Resources 2008; Microsoft 2008). Users dial into the service desk using one standard number, but their call may be routed to any one of a number of locations depending on certain factors such as time of day, local public holidays, call volumes and so on. The advantage of this type of help desk is that there can be 24 hour coverage with each service desk working on normal workday times (Microsoft 2008). This type of service desk, however, shares the same disadvantages of that of a decentralised service desk. Table 2.1 presents the advantages and disadvantages of these service desk structures and it can be noted from the above discussion that they are similar to that of help desks.

The various contact centre types have been discussed and their differences highlighted. Though they seem to be different in terms of functionality, they tend to share the same operating structure which will be discussed in the next section.

2.3 Contact Centre Architecture

Technology within the contact centre helps customers process their transactions quickly and obtain information fast and accurately. This technology is embedded within the architecture of a contact centre to provide effective solutions and maximise customer satisfaction. Figure 2.3 highlights the key components of the contact centre architecture and will be used to understand how they function when a customer calls into a contact centre. This discussion is limited to telephone calls as 84% of all queries are posted by customers are done by telephone (Dimension Data 2006).

A frustrated customer may place a call to a multinational company that he or she purchased a product or service from, by dialling their telephone number. The public service network (PSTN) that provided the company with the number now has access to valuable information about the call: the number of the destination of the call (dialled number identification service [DNIS]) and the number of the origin of the call (also known as the automatic number identification [ANI]) (Gans et al. 2003).

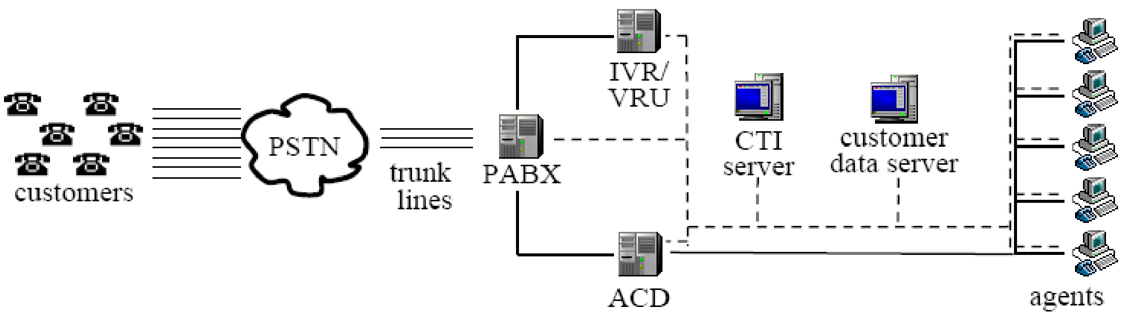

Figure 2.3 Contact Centre Architecture (Gans *et al*. 2003)

The company will have a private automatic branch exchange (PABX or PBX) which will be located by the PSTN through the DNIS. Since the company is multinational the PSTN first has to check the ANI so as to route the call to the closest PABX of the relevant company. The PABX is connected to the PSTN through a number of trunk lines. If one of these trunk lines is free the caller will be connected to the company otherwise the caller receives a busy signal (Bergevin et al. 2008). Calls are usually routed to the Interactive Voice Response (IVR) first before a contact centre agent answers the phone call. IVR is technology that automates interaction with customers through the use of touch tone as a mode of input (Voxeo 2007). IVR ensures that the contact centre agent is not required to answer standard queries and hence they are free to take care of difficult customers or issues that require specific individual attention.

If customers are not able to solve their problem or they wish to speak to a contact centre agent, they are routed from the IVR to an Automatic Call Distributor (ACD), which is a system that can be used to answer, recognise and direct calls to a specific agent (Calltrol 2008). A customer who has called into the contact centre before must be able to speak to the same agent again and the ACD ensures that this happens. Once connected to the customer, contact centre agents can access their computer terminal in order to obtain information on the user and the nature of the problem that they wish to solve.

Companies may also have another form of automation, but this time for the contact centre agents known as Computer Telephony Integration (CTI), which is technology that allows for a telephone and a computer to integrate their functionality (Almskog & Frydman Communications 1996). CTI offers two main functions (techFAQ 2008):

- The telephone system can be controlled by the agent using a computer

- The computer can use the telephone system to display critical information to the agent such as client history, nature of the problem and customer interactions with the IVR. This process is done through the integration of the telephone system to the Customer Relationship Management (CRM) server.

This research focuses on IVR, which will be discussed in detail in the next section.

2.4 Interactive Voice Response

IVR is seen as the most prevalent technology in a contact centre (Dawson 2003; Purushothaman 2004). IVR systems have the ability to retrieve information from enterprise databases according to callers' touch tone inputs. This allows for the customer to perform self-service and access the required data, or route the call to a particular agent group that can handle the specific nature of the problem (Bates and Gregory 2001).

Figure 2.4 illustrates the call flow of an IVR. A customer who calls in and is directed into IVR will encounter an introduction and a greeting (is used to identify the company to the customer). The customer is then presented the first menu, (which is a very broad area in which the customer's problem may be located) from which they must make a selection. If an invalid entry is made by the customer, they will be either directed to a contact centre agent or will be presented with the menu again. If the customer selects a valid menu option, then a sub-menu pertaining to that choice will be presented and this process will continue until the customer solves the problem. If the user presents an invalid input while they are in one of the sub-menus then they are directed to a specialised contact centre agent (an agent who is very knowledgeable on the problem that the customer is trying to solve).

The initial setup costs for an IVR system is considerable, but savings in personnel costs could result in a payback on investment within one business year (Bates *et al.* 2001). Initially there were no choices as to how an organisation could implement IVR. Companies had to somehow tailor their automation solutions to pre-packaged solutions offered by IVR development companies (Dawson 2003). Today there are application generators that could be utilised to tailor IVR to business needs and domain. Application generators also reduce reliance on the ACD unit in order to function and therefore IVR would be tailored to the business needs and wants.

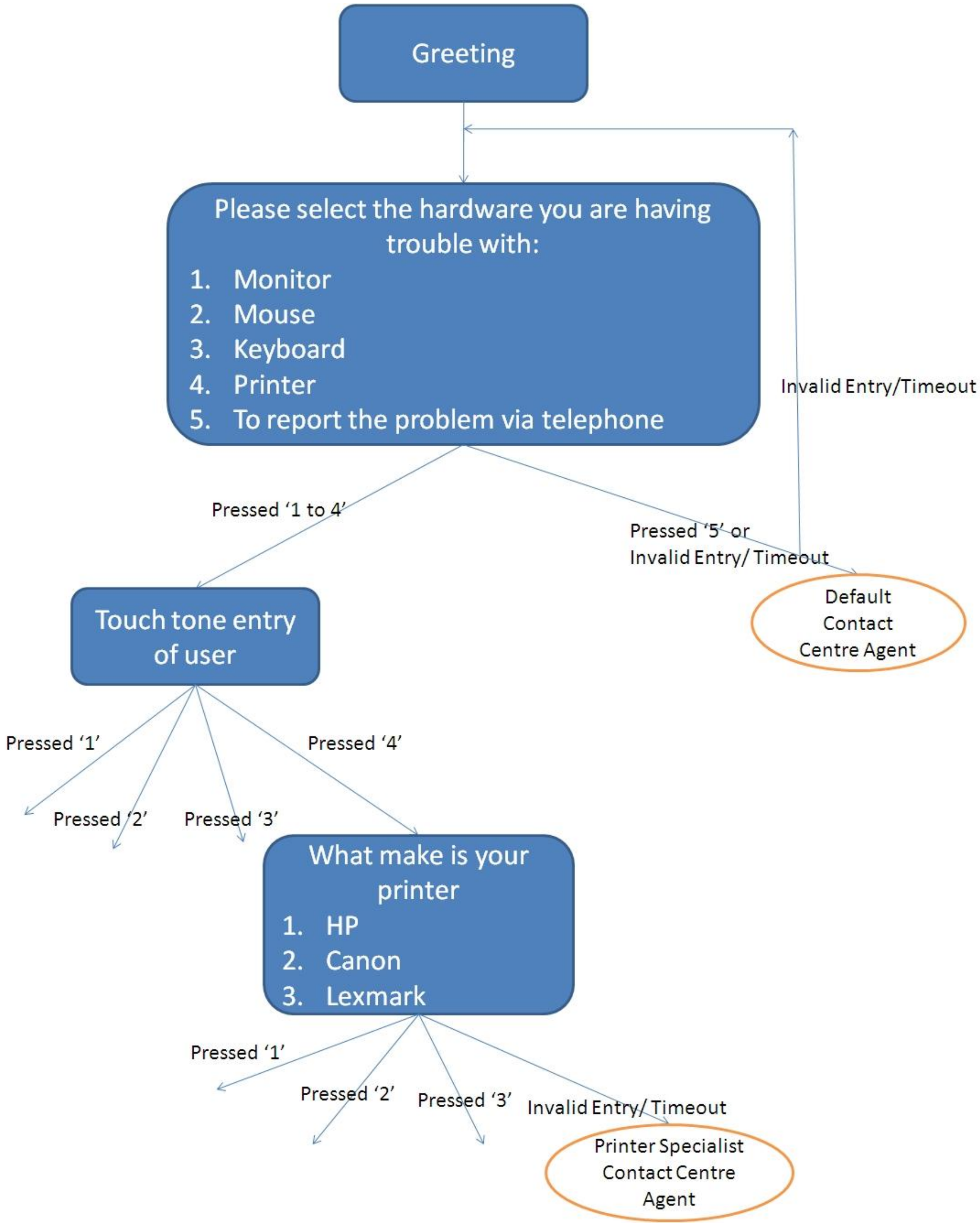

Figure 2.4 Basic Flow of Touch Tone IVR

IVR has four main functions (Purushothaman 2004):

- routing of calls to the appropriate contact centre agent by analysing touch tone input from a customer;

- identify callers and this information can be forwarded to the CTI, which would utilise this information to assist the contact centre agent;
- identify the premium clients and therefore make sure that these clients are serviced first; and
- provide an alternative self-service option for clients willing to solve their problems without having to talk to a contact centre agent.

IVR has a variety of benefits. The telephone system has been present for over 30 years and almost everyone knows how to use it. Customers can access information from anywhere in the world through IVR using a telephone. If your bank makes use of an IVR system to check balances, customers can dial into the IVR system from anywhere in the world to find out their balance at that point in time. It also reduces the need for contact centre agents, as customers would be able to service their own needs. Customers would then only make use of agents to solve complicated queries which cannot be done using the IVR. IVR can reduce 20% to 60% of calls directed at contact centre agents (Dawson 2003).

IVR can be used as a marketing tool when customers are waiting to speak to a contact centre agent. If a customer enters into IVR and has not been able to solve the problem at hand, they will be directed to contact centre agent. If no agents are free, various products and services offered by the company can be advertised to the customer.

Though IVR may seem advantageous due to reduction in call durations with contact centre agents (therefore leading to reduction in costs for contact centres) they have come under scrutiny for two main reasons: the poor interactions with customers and low resolution rates of queries (Saluja 2006). The poor interaction is due to customers finding the menu structure very tedious and difficult to navigate. This was due to the variety of menu options presented to customers (Larson 2005).The menu structure is also not aligned to caller usage patterns and therefore customers have to wait for the correct option that they wish to choose before selecting it (Saluja 2006). New items on the list are usually the last on the list. If this option is related to a new product or service offered by a company, this would not be a good marketing strategy as customers would get frustrated before they hear the new option that they seek (Saluja 2006). Customers are also confronted with unnecessary and irrelevant information in the form of marketing propaganda directed at them and users could have an information overload and could feel flustered (The Ascent Group 2006). These factors have led to a low resolution of queries which has led to low customer satisfaction and retention.

IVR systems have too many deep menus, therefore increasing the interaction times with customers, which as result leaves them frustrated (Dawson 2003). Customers do not get notified of new menu items found in the IVR and as a result do not utilise the service to its fullest potential (Voxeo 2007). When items on the menu are shuffled customers can become confused and are not able to solve their problems.

These problems are concerning, keeping in mind that, for example, Dimension Data (2006) reported that 66% of contact centres responded to calls outside operating hours either by taking messages by IVR, voicemail or answer phone. Customers come into contact with IVRs on a daily basis, unfortunately this has not increased the usability of these systems (Suhm, Bers, McCarthy, Freeman, Getty, Godfrey and Peterson 2002). The abandonment rate of IVR is 12% and the average time of abandonment in IVR is 65 seconds (Dimension Data 2006). Customers have to deal with difficult to use touch tone menus and therefore dislike these systems and seek contact centre agent help at the first opportunity.

In an attempt to address the problems with IVR, contact centres started adopting speech enabled IVRs. These systems however are implemented in an ad hoc fashion and therefore only change the mode of input (speech) and do not increase the usability of these systems.

The next section will address the challenges that contact centres face on a daily basis in terms of automation. Specific attention will be paid to IVR as this discussion lays the foundation for the implementation of automation prototypes.

2.5 Challenges regarding Contact Centres

Contact centres are seen as an organisational tool that facilitates the building and maintenance of the relationship between a company and its customers. Therefore a successful contact centre is seen as one that can maintain loyalty and satisfaction of customers with keeping the operational costs optimal (Singh 2007).

Unfortunately, contact centres have historically had a trend of providing customer service at the lowest possible cost to the company (Siebel Systems 2005). This thinking has lead to their downfall as automation techniques are not aligned to customer needs. Customer satisfaction and retention had therefore suffered. Because of a high turnover rate for contact centre agents, organisations are struggling to provide customers with services at an optimal level. Furthermore companies are faced with inflated costs for both labour and technology (Fusion 2004).

The next three sections give further insight into customer satisfaction, cost reductions and the tradeoffs between customer satisfaction and cost reduction that contact centres face on a daily basis.

2.5.1 Customer Satisfaction

Corporate brands are becoming less important and costs associated with services and products offered by the companies becoming more or less the same, consequently the need for customer service is seen as being absolutely critical and in some industries becoming a deciding factor for clients (Siebel Systems 2005). Every customer transaction has the ability to either strengthen or weaken (and therefore customers them towards a competitor) the relationship between the contact centre and the customer. Contact centres therefore have to play a balancing act between maintaining good customer service and reducing costs. This poses a problem as technology associated with running an efficient contact centre is expensive. Contact centres need to achieve the following goals in order for them to be effective (Siebel Systems 2005):

- a deep understanding of customers needs, wants and behaviour;
- cost-effective solutions must be provided to customers; and
- access must be provide to a well-trained contact centre agent who can solve the customers problem effectively and efficiently when needed.

In order for contact centres to achieve a great sense of satisfaction, various aspects regarding the customer need to be considered (Siebel Systems 2005):

- Contact history of the customer;
- anticipating the needs of the customer;
- customer queries must be handled efficiently and effectively; and
- the contact centre should be able to achieve all of the above in a single interaction.

The variety in communication channels presented by contact centres have made it possible for them to address the various concerns mentioned above.

2.5.2 Cost Reduction

Cost reduction is one of the main aims of a contact centre. Cost reductions have been achieved at contact centres by avoiding the frequent dispatch of costly field staff, the introduction of automation techniques such as IVR and increasing the variety in communication channels (e.g. instant messaging and email) that are cheaper than using the telephone (Siebel Systems 2005; Singh 2007).

Contact centres have also shifted their operating models towards a more outsourcing one so as to reduce costs (Dimension Data 2006). In an outsourcing model all components of the contact centre are leased from an external service provider. In the past outsourcing was seen as the best model as it reduced the costs dramatically. As more and more businesses have opened up trading in the same environment, customers have the choice of easily switching their business elsewhere when service is not up to their standards (Dimension Data 2006). Contact centres have therefore had to find new ways to save costs.

2.5.3 Trade-offs between Customer Satisfaction and Cost Reduction

The tradeoffs between customer satisfaction and cost reduction present contact centres with a dilemma. Quality contact centre agent support is the most effective, but is also the most expensive. The cost of training contact centre agents and the costs of the various technologies to assist them are extremely high. Contact centres have tried to outsource the contact centre agent needs to cheaper economies only to be faced with cultural and language problems (Lester, Branting and Mott 2004).

Automation techniques such as IVR, on the other hand, are very expensive to set up but these systems tend to pay for themselves within a short period of time. These systems however, are difficult to use and tend to lead to low resolution rates of queries posted by users. The choice for contact centres is a difficult one. If an automation technique that was more effective, efficient and satisfying than IVR was implemented, it could be a solution that contact centres could be seeking. This fact highlights the relevance of the current study.

2.6 Conclusions

This chapter highlighted that customer service is key at contact centres for customer retention and attracting new customers. Contact centres have in the past hired as many personnel as they could to service customers; however, this increased their personnel costs.

IVR was introduced to combat increased costs, but, has come under scrutiny. The main problems with IVR that were identified are poor customer interaction resulting in a low resolution of queries resulting in low customer satisfaction. Companies therefore need to address the trade-off challenges between customer satisfaction and cost reduction.

Within this scenario an investigation into improved automation techniques become very relevant. An overview of rule-based expert systems (Chapter 3) and natural language interfaces (Chapter 4) forms an important part of this investigation.

Chapter 3 – Rule-Based Expert Systems

It is not my aim to surprise or shock you--but the simplest way I can summarize is to say that there are now in the world machines that can think, that can learn and that can create. Moreover, their ability to do these things is going to increase rapidly until--in a visible future--the range of problems they can handle will be coextensive with the range to which the human mind has been applied. --Herbert Simon

3.1 Introduction

The birth of Artificial Intelligence (AI) happened in 1956 at Dartmouth College during a summer workshop sponsored by IBM (Durkin 1994). The workshop considered different techniques that could be utilised in automatic theorem solving. This brought about theories on how simulation of human reasoning could be implemented. Expert systems were one option that was highlighted. Figure 3.1 highlights the various areas of interest of AI and the three areas that are discussed in this research namely: *expert systems*, *natural language understanding* and *speech recognition and understanding*. The area of *expert systems* highlights a technique for simulating the way in which humans reason and draw conclusions from facts. *Natural language understanding* is used to understand input that is given in natural language. This research will discuss pattern matching as a natural language understanding technique in Chapter 4. *Speech recognition and synthesis* are not the focus of this research but will be discussed briefly as a mode of natural language interaction in Chapter 4.

A Rule-Based Expert System (RBS) can be defined as "an intelligent computer program that uses knowledge and inference procedures to solve problems which are difficult enough to require significant human expertise for their solution" (Feigenbaum 1982). RBSs have been utilised in various business domains such as aeronautics, medicine and science. RBSs utilise specialised knowledge to solve problems at the level of a human expert.

Computer programs that use conventional problem-solving methods make use of algorithms, data structures and reasoning in order to find solutions (Badiru and Cheung 2002; Abraham 2005). For more difficult problems that require the need to simulate human experts (utilising human reason and draw conclusions from facts) RBSs constitute a good means of codifying the know-how of a human expert (Durkin 1994).

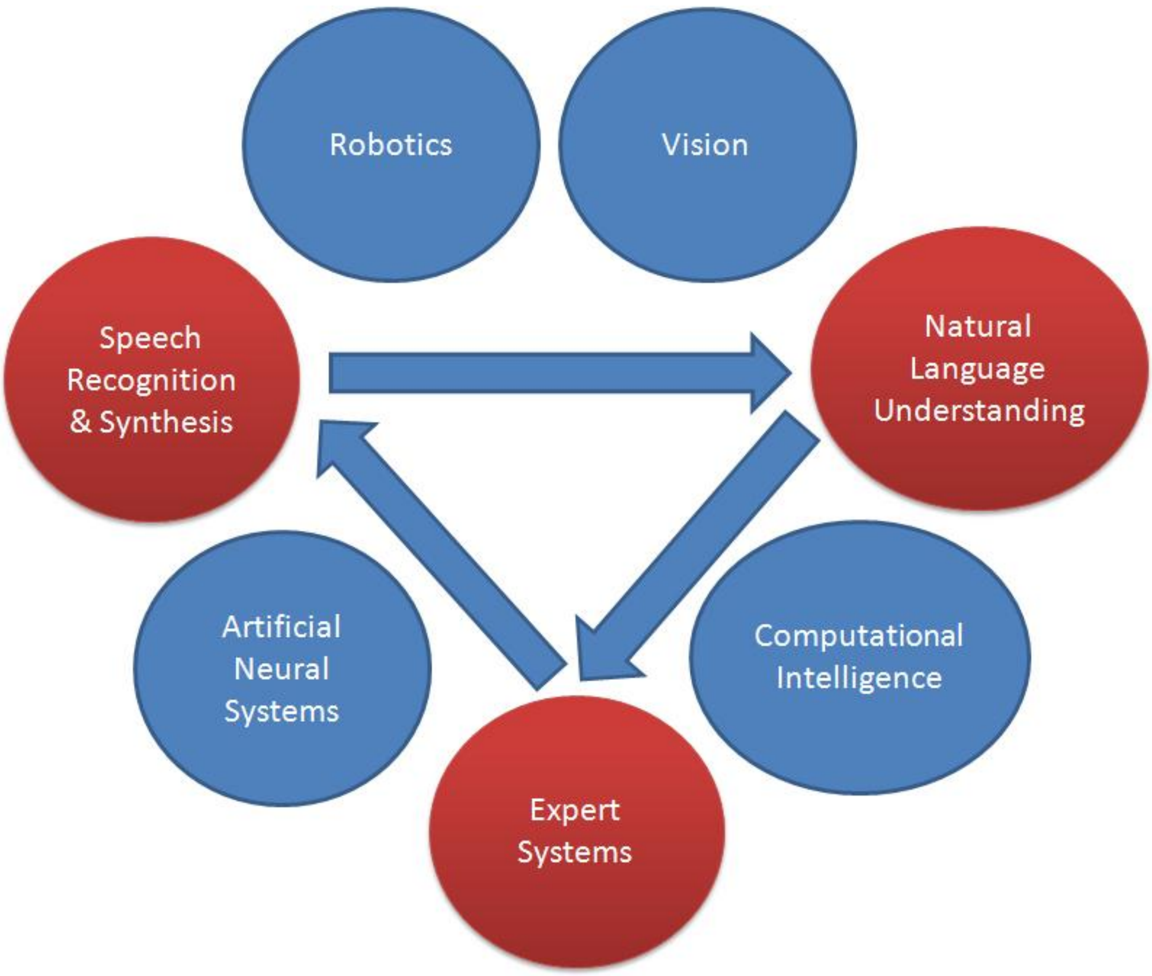

Figure 3.1 Some Areas of Artificial Intelligence (Giarrantano *et al.* 1989)

An expert is defined as a person who has vast experience in a field (Giarrantano *et al*. 1989). Experts express their actions while problem solving through situation-action rules and this is the same way that RBSs function. Knowledge stored in the knowledge base of an expert system are mostly rules-of-thumb or heuristics, although they may also contain well-proven rules. Rules are also known as productions and generally take this form:

If *condition* ***then*** *action*

The premise (***if*** part of the rule) and the conclusion (***then*** part of the rule) are sometimes given a certainty value. This certainty value refers to the probability of the conclusion that is reached being true. The premise is a set of restricted sequence of clauses connected by the connectives *and* or *or*.

Knowledge found in these systems may either be expert or generally available from books and magazines. The person performing the task of acquiring knowledge is known as a knowledge engineer and the process of collecting and structuring knowledge is known as knowledge acquisition. RBS are, therefore, also referred to as knowledge-based or knowledge-based expert systems.

Many different techniques have emerged for organising collections of rules into automated experts, but despite this all RBSs share certain key properties (Hayes-Roth 1985):

- practical human knowledge is incorporated through conditional if-then rules;
- the larger the knowledge base the higher the skill;
- a wide range of problems can be solved by selecting the right rules and then combining the results in the appropriate way;
- the best sequence of rules to executed is done adaptively; and,
- Conclusions are explained by retracing their actual lines of code and translating each rule into natural language.

The objective of this chapter is to identify the benefits and shortcomings of RBSs. Their architecture is discussed to understand how the implementation of these systems can be conducted. The different types of RBSs are also discussed with their benefits and shortcomings presented.

3.2 Rule-Based Expert Systems

A RBS is a knowledge-based system which is dedicated to specific tasks requiring a great deal of knowledge or expertise about a particular domain. The expertise found in an expert system is permanent and can be used to simulate scarce or expensive human experts (Grzymala-Busse 1991). Domains that have only a few and costly experts who have an incomplete theoretical foundation can benefit from the use of expert systems. If the data or the ways human experts solve their problems are not explicitly defined expert systems may prove to be the most practical implementation (Grzymala-Busse 1991). With problems that need creative thinking, imagination or common sense people have an advantage over expert systems.

RBSs have been implemented in a variety of domains but they generally fall into one of the following categories (Badiru *et al.* 2002):

- interpreting and identifying;
- predicting;
- diagnosing;
- designing, planning and monitoring;
- instructing and training; and
- controlling.

RBSs have been found to perform as well as human experts or in some cases even to outperform them. This is mainly due to two reasons (Forsyth and Naylor 1991):

- When RBSs are implemented they always provide consistent solutions because the system does not forget nor have an "off" day; and
- Furthermore the designing and production of an RBS requires one to really analyze that domain in detail. When a domain is analysed to its full extent, it creates a new repository of knowledge which could lead to an overall improvement in human expertise.

Some of the drawbacks of RBSs are that it is not possible to implement a solution for every domain. Some problems are very human specific. An example of this would be someone who has fractured a leg. This would require a doctor to examine the leg visually.

This would not be possible with an RBS. The second and more important problem is brought about by the knowledge engineer; if the knowledge engineer makes some mistakes when creating the premises and conclusions this could lead to the system giving misleading and incorrect information to a user.

RBSs have been implemented in a variety of domains. The architecture of an RBS is discussed next which will lead to a discussion on the various types of inferences an RBS can make.

3.3 Rule-Based Expert System Architecture

For an RBS to make decisions, it needs its knowledge to be structured in an efficient manner in order for the computer to retrieve the information in an efficient and effective manner. The knowledge must be organised in such a way that it is easily accessible and that there is a distinction between data, knowledge and control structures (Badiru *et al.* 2002). This fact has lead expert systems to be organised into three distinct levels: a *knowledge base,* a *working memory* and an *inference engine* (Figure 3.2).

Knowledge-based RBSs collect knowledge in the form of human know-how in a *knowledge base*, which is utilised to reason through a problem in order to find a solution. The knowledge base is one of the most important components of a RBS, as without it the system would not be able to function. The knowledge base can combine the knowledge of multiple human experts. Knowledge is created through the use of rules and facts. A rule is a conditional statement that links given conditions to actions or outcomes (Petrovic 2000). Facts represent what we know at any given point in time.

The *working memory* refers to task-specific data for the problem under consideration. This memory is used to gather all the data the system possesses about the problem it is currently addressing.

The purpose of the *inference engine* is to retrieve information and relationships from the knowledge base and provide answers, suggestions and tips as a human expert would (Hayes-Roth 1985). Inferences can occur in two main types namely forward chaining or backward chaining (Lucas and Gaag 1991), and will be discussed in next section. Facts, rules and interpretations must be assembled correctly for the inference engine to function properly.

There are three other optional components found in RBSs which are the knowledge-base acquisition facility, explanation facility and the user interface (Figure 3.2).

The explanation facility is present to allow users to understand how the system arrived at a certain result. The knowledge-base acquisition facility is used to provide an easy way to capture and store information in the knowledge base (Abraham 2005). The user interface is utilised to interact with the user (to find out more information from the user and to display information to the user).

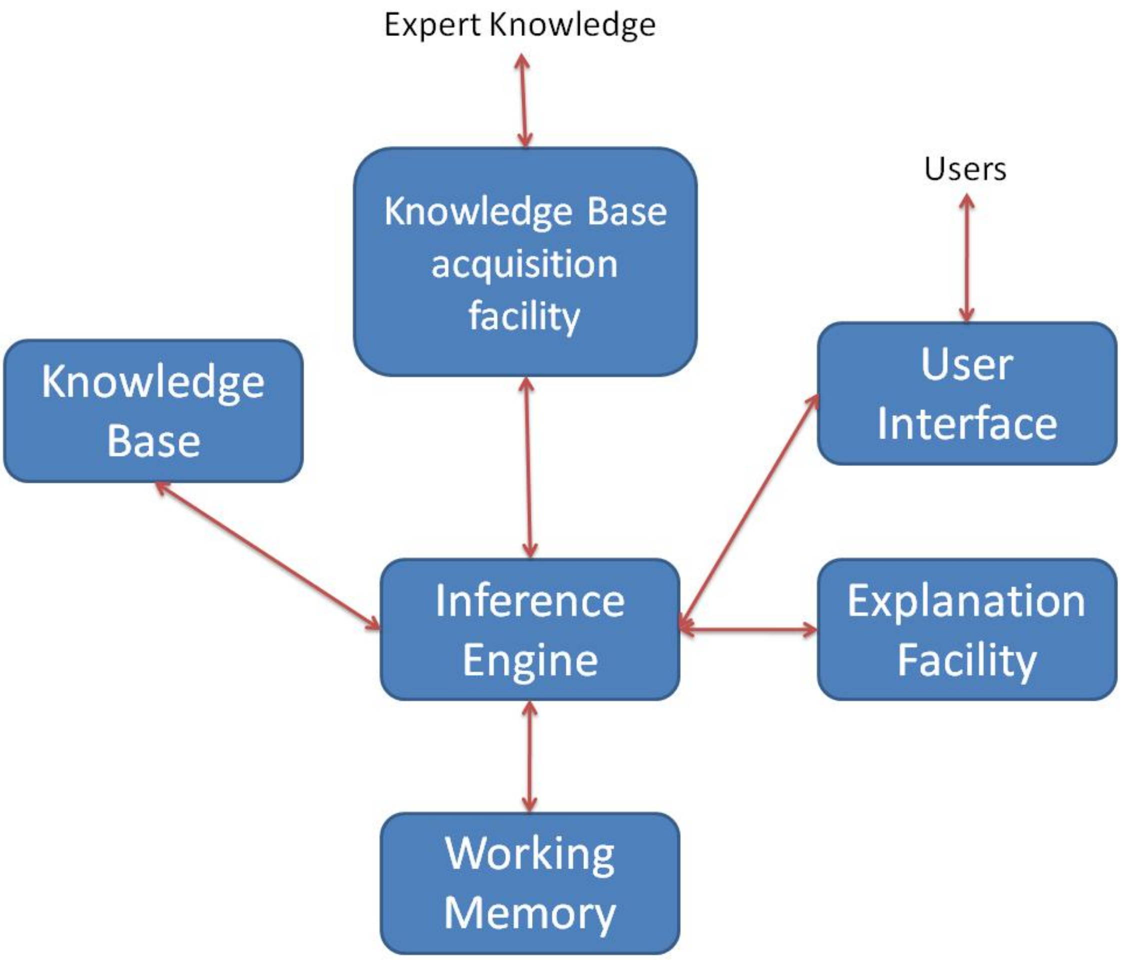

Figure 3.2 Architecture of a Simple RBS (Abraham 2005)

3.3.1 Inference in Rule-Based Expert Systems

Knowledge stored in the knowledge base of an expert system is mostly rules-of-thumb or heuristics, although they may also contain well-proven rules. Rules are also known as productions and generally take this form:

If condition ***then*** action

The above condition can take the form '*if* x is A *then* y is B'. The *if* part of the statement is known as the antecedent or premise, while the *then* part of the rule is called the consequent or conclusion. Figure 3.3 highlights some of the rules of a printer troubleshooting RBS. There are two types of inferences that exist namely forward chaining (data-driven) and backward chaining (goal-driven).

If the printer is not printing ***then*** find out if it is on

If the printer is not printing ***and*** it is on ***then*** see if there is paper

If the printer is not printing ***and*** it is on ***and*** there is paper ***then*** check for a paper jam

Figure 3.3 Representation of a Printer Troubleshooting RBS Rules (Hayes-Roth 1985)

3.3.1.1 Forward-Chaining

One of the possible methods of inference is known as forward chaining which is a data-driven method. Forward chaining works in the same way that any person might reason in trying to establish as much information about a problem as possible with only a few facts known about the problem.

Inference procedures start with a set of known facts contained in the working memory and these are utilised to derive new facts using rules which match with the known facts. When the inference engine has multiple rules that match the known set of facts, a rule is chosen randomly by the inference engine. This process continues until the goal has been reached or when there are no more rules that can be matched utilising the facts in working memory. These are either the original facts or those derived during the process.

Consider the following rule base:

Rule 1: ***If*** the printer is on ***and*** the page is blank ***then*** replace toner

Rule 2: ***If*** the printer is on ***and*** there is paper ***then*** it is a paper jam

Rule 3: ***If*** the red light is flashing ***then*** there is a paper jam

Rule 4: ***If*** there is a paper jam ***then*** look for any stuck paper in the printer

The user of the RBS has noticed that there is a red light flashing (and assumes the printer is on due to this fact) on the printer and the document has not printed. Using forward chaining the system will analyse Rule 1 and then go downwards until a rule that fires is found. If we go downwards, only Rule 3 would fire given the set of known facts (printer is on and a red light is flashing). It has now been deduced that there is a paper jam and this would be used as input for Rule 4 which would tell the user of the RBS to look for paper stuck within the printer causing the paper jam.

The major advantage of a forward chaining system is that only a small amount of data is required, which could be utilised to derive a lot of information.(Lucas *et al.* 1991). The major disadvantage of these systems is that they have no way of determining which rules are more important and have a greater impact on an outcome.

A forward chaining strategy is especially appropriate in situations where data is expensive to collect, but small in quantity. Forward chaining uses bottom-up reasoning by trying to find solutions that can be found from the facts. These types of systems are data driven and therefore need interaction from the user in order to function.

3.3.1.2 Backward Chaining

Backward chaining unlike forward chaining tries to obtain as much information as possible from known facts. Backward chaining tries to establish information that will lead to the proof of a given hypothesis (Lucas *et al*. 1991). The process starts by obtaining the goal that must be proved and then checks the working memory whether this goal fact has been added previously or not. If the goal fact has not been added it searches through all the premises to find those rules which have the goal as a consequence after its conclusion (Kandel 1991).

Special rules, known as goal rules, are utilised, which the inference engine tries to make true. The goal rules are checked to see if they are listed in the working memory. If the goal is found, then it is assumed to have been proved if not, the premises that have not been listed will now become sub-goals (Forsyth *et al.* 1991).

The above process occurs recursively until the system encounters a premise that is not concluded by any rule. This type of proposition is known as a primitive (Kandel 1991). When a primitive value is encountered, the system queries the user through a user interface for a truth value. The primitives are then utilised to prove sub-goals or the original goal.

Consider an example utilising the same rules as in the previous section:

Rule 1: ***If*** the printer is on ***and*** the page is blank ***then*** replace toner

Rule 2: ***If*** the printer is on ***and*** there is paper ***then*** it is a paper jam

Rule 3: ***If*** the red light is flashing ***then*** there is a paper jam

Rule 4: ***if*** there is a paper jam ***then*** look for any stuck paper in the printer

When using the backward chaining method, the process starts with the desired goal and attempts to prove that goal. If the desired goal is to prove that the *user must look for paper stuck in the printer* (Rule 4) given that the *printer is on* and *there is a red light flashing*, then the task starts by looking for a rule that proves that the *user must locate any paper stuck in the printer*. Rule 4 proves that this is true if there is a *paper jam*. Now Rule 2 comes into play, and it is already known that the *printer is on* and that *there is a red light flashing*. Given that the condition *the printer is on* is found in Rule 2, the new sub goal is to prove that there is paper in the printer. Rule 3 proves that there is paper in the printer given that there is a paper jam and therefore it can be concluded that there is a paper jam which proves that the user must look for any paper that is stuck in the printer.

Backward chaining systems, in contrast to forward chaining systems, perform very well in problems on which a hypothesis is stated and then tested. The goal remains the focus and only relevant information needed by the systems will be derived from the user. The major disadvantage of backward chaining systems is that they tend to pursue lines of questioning which should be dropped while others should be investigated (Kandel 1991). The single-mindedness of these systems, therefore, allow for more computation than is necessary as entire branches of useless properties are visited before backtracking can occur (Vogts 1998).

Backward chaining is useful in situations where the quantity of data is potentially very large and where some specific characteristic of the system under consideration is of interest. The system reaches conclusions by evaluating every possible solution by using top-down reasoning and would be useful for diagnosis of problems especially in the medical field.

Table 3.1 summarises the differences between forward chaining and backward chaining RBSs. A forward chaining strategy is very similar to the way IVR functions in contact centres. Users input information which is used by the system to reach a conclusion on the nature of the problem and a solution is provided. In this documentation reference to IVR will always imply a forward chaining RBS.

Forward Chaining	Backward Chaining
Planning, monitoring, control	Diagnosis
Present to Future	Future to present
Data Driven, bottom-up reasoning	Goal Driven, top-down reasoning
Work Forward to find what solutions follow from the facts.	Work backward to find facts that support the hypothesis.

Table 3.1 Characteristics of Forward and Backward Chaining

3.4 Advantages and Disadvantages of Rule-Based Expert Systems

The major advantage of an RBS is that it provides consistent answers to repetitive questions and can hold and maintain significant levels of information (Welsh 2006). It can also reduce employee training costs as these systems are relatively easy to use. Customers would be able to utilise them easily. Furthermore multiple expert intelligence can be combined, further reducing costs associated with hiring experts. Expert systems also have the ability to store previous responses unlike humans who may forget that customers have already volunteered the information.

The major disadvantage of RBS is, however, its lack of common sense which a human possesses while making decisions. Furthermore, an expert system may not be able to come up with creative answers to queries, as humans could do in unusual circumstances (Welsh 2006). And finally, unlike humans, expert systems cannot recognise when they do not have an answer to a query in the earlier stages of problem solving unlike humans.

The advantages and disadvantages mentioned above are also relevant to IVR systems as they are an example of a forward chaining RBS.

Table 3.2 summarises the advantages and disadvantages of expert systems.

Advantages	Disadvantages
Consistent answers provided for repetitive decisions	Human common sense is lacking in some decision making
Reduces the amount of human errors	Will not be able to give creative answers like that of human experts in unusual circumstances
Holds and maintains significant levels of information	Not being able to recognise when no answer is available for a specific problem
Reduce employee training and personnel costs	Cannot adapt to flexible environments.

Table 3.2 Summary of Advantages and Disadvantages of RBS

3.5 Conclusions

The objective of this chapter was to introduce rule-based expert systems. This was achieved through the definition of an RBS, the architecture that these systems use and the two inference types that they utilise. The advantages and disadvantages of these systems were also discussed.

RBSs consist of three main components which are the knowledge base, the working memory and the inference engine. These components work in tandem in order to reach a solution to a problem. The main problem with an RBS is that the knowledge base has to be carefully constructed by the knowledge engineer because if mistakes are made they could lead to wrong conclusions being made.

RBSs are a very interesting field of AI and are extremely useful in the appropriate domain. They have been utilised successfully in various business domains such as aeronautics, medicine and contact centres. Contact centres have utilised RBSs for their automation purposes by applying the forward chaining strategy in implementing IVR systems. The advantages and disadvantages of RBS that were discussed are also relevant to IVR systems.

The next chapter introduces natural language interfaces as an alternative approach to automating human computer interaction.

Chapter 4 - Natural Language Interfaces

The newest computer can merely compound, at speed, the oldest problem in the relations between human beings, and in the end the communicator will be confronted with the old problem, of what to say and how to say it. -- Edward R. Murrow

4.1 Introduction

Human beings are very unique creatures as we have the ability to learn and use natural language. This ability allows humans to have natural, efficient and intelligent conversations with each other. Humans currently interact with computers (human computer interaction) through the use of various hardware devices such as the mouse, keyboard and monitor in addition to software devices such as menus and dialogue boxes. These are, however, not comparable to human-to-human conversations. Computers are increasingly becoming ubiquitous in nature, therefore, there is a need for a more natural interaction with these devices. One method that could be utilised is through the use of a more human-human technique such as natural language. This would provide the user with the sense that the system is behaving intelligently, and would invoke the feeling that they have personalised interaction with the system (Liberman 2001).

Natural language can be processed by computers through *speech recognition, speech synthesis, text-based pattern matching* or *gesture interaction*. *Speech recognition* has come of age in recent years but has not matured to an extent that it can be utilised to have a conversation with a computer (Winograd and Flores 1986; Long 1994). Most speech recognition systems are restricted to certain keywords within a domain and, therefore cannot be utilised for accomplishing tasks outside a particular domain (Dusan and Flanagan 2004). Furthermore, these systems do not adequately cover all utterances a user might utilise in order to accomplish a task in a certain domain. Research in the area of natural language processing is moving towards the creation of systems that are capable of learning human knowledge and obtaining related knowledge as a conversation proceeds (Dusan *et al*. 2004).

Speech synthesis is the process of outputting simulated human speech (Dutoit 1997). Speech synthesis has been more successful than speech recognition and has a variety of applications. Contact centres use speech synthesis in the IVR to present the customer with menus with which they interact. Furthermore feedback is also provided to the customer using speech synthesis.

Text-based pattern matching has mostly been utilised by conversational agents or search engines. These systems utilise pattern matching in order to interact with the user. This chapter will introduce a conversational agent called ALICE, which utilises pattern matching techniques to converse with the user. Pattern matching is the act of checking text for a given structure and if it matches a certain task is performed. In the case of ALICE when a pattern is matched a template will be triggered. This will be discussed in more detail in Section 4.3.2.

Gesture interaction is the process of analysing human gestures. Humans usually gesture when using human language in order to interact with other human beings. These gestures can convey the mood and sometimes the context of a conversation. In some cultures bowing in front of someone is a method of saying hello (therefore the context of the conversation at that point is introduction). Gesture interaction is usually utilised as a complimentary technique to speech (Cassell, Steedman, Pelachaud, Stone, Douville, Prevost and Achorn 1994). They provide a method through which context can be maintained in a conversation. Furthermore, the mood of the conversation can also be understood by the system. Gesture interaction is ***not*** discussed further as it is not the focus of this research.

The objectives of this chapter are to identify the different types of NLIs, their benefits and shortcomings. NLIs are introduced in Section 4.2. Particular attention is paid to a text based pattern matching interface known as ALICE. The various advantages and disadvantages of the ALICE system are also presented (Sections 4.3.3 and 4.3.4). An overview of different NLI applications is given in Section 4.4.

4.2 Natural Language Interface as a User Interface

An NLI is defined as an interface that has the ability to interact with users using human language such as English, as opposed to computer language, a command line interface, or a graphical user interface (Ogden and Bernick 1996; Zhou, Shaikh and Zhang 2004). The interface takes as input either written text or spoken speech. These types of input may be utilised individually or in a combination to produce a multimodal input interface. As stated in Section 1.3.3, this research only focuses on textual input.

NLIs are usually only capable of understanding a restricted subset of a human language (usually restricted to a certain domain) and generate more or less pre-packaged responses (Patridge 1991). The user would have to learn to utilise a small subset of the English language in order to operate most of these interfaces (Cohen, Dalrymple, Tyler, Moran, Pereira, Sullivan and Gargan 1989; Androutsopoulos 2002). Some experts however feel that this is not an effective NLI as users would have to learn how to use the system before being able to utilise it effectively (Ogden *et al.* 1996). They argue that applications that utilise natural language should stimulate conversation between the human user and the computer in order to have a successful communication between the two parties. Human-to-human conversations take place by taking turns between speaker and listener. When NLIs are designed this should therefore be taken into account.

NLIs were first utilised for human-computer interaction through natural language in 1966 with a system, called ELIZA, created by Joseph Weizenbaum. ELIZA was a simple conversational agent (Section 4.2.2) that was capable of parsing simple sentences and utilising them to pick out keywords. These keywords were then utilised for substitutions to turned into questions. ELIZA was not capable of holding a long conversation with a user as most statements made by the user were merely turned into questions. Though ELIZA did not utilise any sophisticated processing techniques, it was still a notable NLI as it was the first time human-computer interaction occurred through natural language. The different modes of input through natural language, that are specific to this research, are discussed next.

4.2.1 Conversational Speech Interfaces

A conversational speech interface is one that utilises speech as an interaction technique and is an example of a natural language interface. This interface operates utilising the following functions: speech recognition (input) and speech synthesis (output). A major benefit of incorporating speech in applications is that it comes naturally to humans. Most people find speaking and listening easy. Though speech is easy for humans it is not so easy for computers. The reason for this is that speech technologies lack 100% accuracy (Lai and Yankelovich 2003), and there is a problem with ambiguity when utilising natural languages (Hendrix 1982).

The main problem when designing these interfaces is that system developers do not design these systems with speech in mind from the beginning but rather in terms of the graphical user interface. This brings about a problem of merely designing a command line interface which utilises speech through a graphical user interface (Lai *et al*. 2003). Another crucial factor in determining whether or not the application will be successful is to determine whether there is a clear benefit to utilising speech (Lai *et al*. 2003). This involves assessing whether speech is absolutely necessary, in other words when the users hand or eyes are busy or when the task to be completed cannot be accomplished without the use of speech. Speech is not suitable in situations such as when large amounts of information need to be presented to the user.

4.2.1.1 Speech Recognition

Speech recognition can be seen as the process by which a computer can identify the parts of human speech (Jelinek 1997). The process starts with the user uttering something into a microphone and ends with the computer accomplishing a task. The solution that has not been successfully implemented by humans for computers is to accurately identify all possible words spoken by any person in any environment (Hill 1983; deWet, deVeth, Cranen and Boves 2003). Systems performance in speech recognition can be affected by a number of factors including large vocabularies, multiple users, continuous speech and noisy environments.

When a user speaks into a microphone, phonemes are extracted from the user speech. Phonemes are the linguistic units of human language (Zue, Cole and Ward 2000; Matthews 2002). These sounds are grouped together to form words utilised in human language. When the sounds are grouped together the actual process of understanding user input is initiated. Over the years, many approaches have been utilised but only two will be discussed here namely pattern matching and knowledge-based approaches. These two approaches are not mutually exclusive and are discussed below:

- Pattern matching - The goal of pattern matching is to take an unknown pattern and compare it to a set of known and stored patterns (also known as templates). These are established through training data. Templates are utilised to compare the pattern and compute a similarity score (Zue et al. 2000; Schroeder 2004). The template with the highest score is then chosen as it will have the highest acoustic similarity to the users input.
- Knowledge-based approaches - This approach makes use of a rule-based expert system which (as mentioned in Chapter 3) utilises a base classification of rules in order to function. However to utilise this model, a large set of rules would be needed in order to capture a great variability in speech (Kaminski 1989). Rules are formed from knowledge about speech signals. This approach could be useful, however, it does not perform effectively if there is insufficient knowledge.

4.2.2.2 Speech Synthesis

Speech synthesis enables computers and other electronic devices to output simulated human speech (Dutoit 1997). A computer system that is utilised for the purpose of producing human speech is known as a speech synthesiser. A text-to-speech (TTS) system is an example of a speech synthesiser that converts normal text into speech (Holmes and Holmes 2001). The quality and effectiveness of speech produced by these system are measured by utilising these characteristics (Karat, Vergo and Nahamoo 2003):

- Base-level achievement of speech that is intelligible (the ease with which the output is understood) to humans;
- produce speech that is as natural as that of human beings in other words how natural the speech is;

- produce speech that is personalised to a particular user, in other words it has the same intonation as a person's speech; and
- the final level and highest level of achievement is to produce speech based on a person's own voice recordings so that the speech sounds as if it belongs to that person.

For a text-to-speech system to function, a narrator is utilised to record a series of text (such as reading from an encyclopaedia, poetry, political news and various other texts). These narrations are carefully picked in order to ensure that every possible sound in a given language is recorded (Acapela Group 2008). These narrations are then sliced into the different phonemes found in the particular language and stored in a database (Figure 4.1). When the database is created, the various recorded utterances are segmented into one of the following: diphones (sound-to-sound transitions), syllables (units of organization for a sequence of speech sounds), morphemes (smallest linguistic units that have semantic meaning), words, phrases and sentences (Dutoit 1997; Acapela Group 2008).

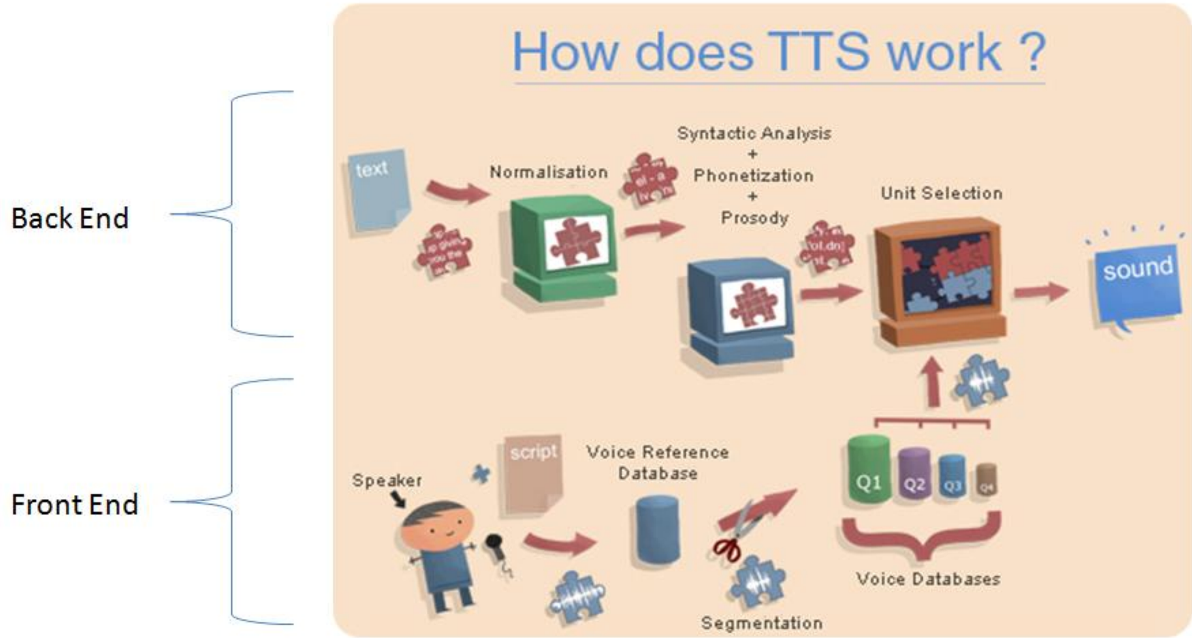

Figure 4.1 How TTS Works (Acapela Group 2008)

The above-mentioned process occurs in the *back end* of the TTS system. The *front end* has two major tasks namely: *normalization* and *text-to-phoneme conversion* (Santen, Sproat, Olive and Hirschberg 1997). *Normalization* is the process that occurs first, where the conversion of raw text occurs. All abbreviations and symbols are expanded into their respective words. These are then passed down to the component that starts the *text-to* -

phoneme conversion. Once the text is passed on this component makes sure that the sentence is syntactically correct. The various phonemes are then extracted for that particular sentence and produced as human speech through a speaker or any other device capable of producing sound (Figure 4.1).

There are two types of speech synthesis that are utilised for commercial applications namely concatenated synthesis and formant synthesis (Karat *et al*. 2003). Concatenated synthesis employs computers to assemble narrators recorded voices into speech signals. Though this sounds fairly simple, it is very database intensive and has large storage needs to store all recorded speech (Karat *et al*. 2003). Formant synthesis, on the other hand, utilises a rule-based expert system. This expert system applies a set of phonological rules to a specified audio waveform which simulates speech.

4.2.2 Text-Based Natural Language Interfaces

Text-based NLIs utilise text instead of speech in order to function. Text may be in the form of a query, sentence or a list of keywords (as used by search engines). A user will type in a question in an appropriate field (usually a textbox) and the system will retrieve information in accordance to the users query. Text-based natural language interfaces have been utilised in many applications including: databases (query and report generation) (Hendrix 1982; Wu, Ichikawa and Cercone 1996), conversational agents and search engines (matching user requests to keywords).

Conversational agents are a communication technology that utilize natural language and various linguistic methods to interact with human users through natural language (Lester *et al.* 2004). Conversational agents need to satisfy two sets of requirements for them to be effective (Lester *et al.* 2004):

1. they must have good language processing capabilities such that they have the ability to engage in productive conversations with the user. This involves understanding user input and employing effective dialogue management techniques; and
2. they must be scalable and reliable and allow for smooth integration within business processes.

Conversational agents are generally deployed on retail websites and are utilised by customers to enquire about products and services. However as these agents gained popularity they were deployed in a variety of other domains such as education (Ventura, Ventura and Olabe 2005), banking, travelling agencies and a variety of others (McBreen, Anderson and Jack 2000). Conversational agents have also been utilised in the virtual world and because these worlds are mostly text based in nature, they make an ideal environment for communication with the user. Figure 4.2 depicts a personal assistant that utilises conversational agent technology in order to interact with the user. These types of virtual conversational agents can be found on social websites such as Secondlife (Lester *et al.* 2004).

Another conversational agent known as ALICE utilises simple pattern-matching techniques to engage users in a conversation. ALICE utilises Artificial Mark-up Language (AIML), which is a derivative of XML to store its patterns and templates. Section 4.3 provides a detailed discussion on ALICE.

Figure 4.2 A Virtual World Personal Assistant

Pattern matching is not the only technique utilised, various other methods such as Bayesian networks are also used. There are mainly two methods utilised by text-based interfaces to comprehend user input namely:

- *Pattern matching* – conversations and their responses are stored in pairs. The pattern is the user input or stimulus which is matched to produce a template which is the output of the system. The pattern can be seen as a simple text string that has to match the input exactly. The template is the output that a user receives as a response (as it was entered by the person who created the conversational agent). The biggest flaw with pattern matching is that it is difficult to maintain context of a conversation. Moreover it is impossible to cater for all possible inputs.

- *Bayesian networks* - is a graphical model that represents a set of variables and their probabilistic dependencies (Murphy 1998). It is implemented in conversational agents to maintain context in a conversation. Context can be maintained through Bayesian networks as the probability of the next question can be calculated. This probability is calculated by analysing current and previous user input. The question with the highest probability is usually always chosen.

The scope of this research is restricted to conversational agents with special emphasis on ALICE (Section 4.3).

4.2.3 Shortcomings of Natural Language Interfaces

Human-computer interaction experts believe that NLIs are not as attractive as they initially appeared to be (Long 1994). Early literature in this field focuses on the shortcomings of these interfaces in terms of user task completion (Fum, Guida and Tasso 1988; Cohen 1992). The shortcomings are brought upon by the ambiguous nature of a natural language and heavy dependence on a huge repository of knowledge (Fum *et al*. 1988).

Consider the following quote:

At last, a computer that understands you like your mother – **1985 McDonnell-Douglas ad**.

A computer can interpret this quote in a number of different ways (ambiguity) and thus will show us the difficulties in analyzing human language. If we look at the sentence carefully there are three interpretations that are possible (Lee 2004):

- the computer understands you as well as you mother understands you;
- the computer understands that you like your mother; and
- the computer understands you as well as it understands your mother.

This shows us how a simple sentence can be ambiguous for a computer, while we as humans can easily rule out all other alternatives except the first one. We do so, based on a great deal of background knowledge, including understanding what advertisements typically try to convince us of (Lee 2004).

There are various techniques that could be used to get around this flaw. One of the oldest techniques used is to *limit the syntax and vocabulary* that exists in a natural language (Long 1994).

Another technique that could be used is the *use of certain linguistic theories*, which outline certain rules that should be followed while conversing. Examples of such rules include:

- users should not give more information than necessary;
- users should not make unnecessary speech contributions than is necessary; and
- users should not intentionally make ambiguous references but rather use references that they believe will unambiguously describe exactly what they what want to achieve.

The most popular technique used to combat ambiguity is to *engage in some form of clarification or confirmation dialogue* to confirm if the interpretation is, in fact, the correct one (Cohen 1992).

4.2.4 Benefits of Natural Language Interfaces

Though NLIs have their disadvantages, they also have a variety of advantages. The main reason why NLIs are seen as beneficial is because no prior training is required in order to utilise them as people use natural language in order to interact with each other on a daily basis. Other benefits of conversational speech interfaces include (Hill 1971; Schroeder 2004; Zhou *et al.* 2004):

- Offers natural communication;
- Allows for physical mobility when troubleshooting problems that are not near the interface; and
- Allows for flattening of deep menu structures found in some computer systems. This would be possible as users could state their queries in natural language and would eliminate the need for menus. This would be beneficial for systems such as IVR.

4.3 ALICE

ALICE is the "Artificial Linguistic Internet Computer Entity" implemented by Dr. Wallace in 1995 (Sharwar and Atwell 2007). The ALICE knowledge base makes use of conversational patterns which are stored in Artificial Intelligence Mark-up Language (AIML) files and this will be discussed in Section 4.3.1. AIML is used by the pattern matching algorithm in order to provide an output for user input (Section 4.3.2). Since ALICE forms an integral part of this research, it is important to focus on its strengths (Section 4.3.3) and weaknesses (Section 4.3.4).

4.3.1 Artificial Mark-up Language (AIML)

AIML is a mark-up language which is utilised to match a sequence of words to generate a response (Wallace 2005). Responses are stored templates and can be utilised for the substitution of input words into an output template. AIML is an open-source standard and has therefore gained tremendous popularity with the general public. Dr. Richard Wallace (the founder of ALICE) created AIML with the aim of allowing people who could create webpages to be able to create a conversational agent.

In AIML, each stimulus/response pair is stored in a category and this is the basic unit of knowledge. Each category consists of two elements known as the pattern and the template. A pattern is the user input, or stimulus, and the template is the response, or output (Figure 4.3). The ALICE brain contains hundreds or thousands of AIML categories. The pattern can be seen as a simple text string that has to match the input exactly, while the template is the output that the user receives as a response as it was entered by the person who created the conversational agent.

It would be difficult, however, to cater for every possible input that a user will input into the system. AIML solves this problem by utilising wildcards. As in the Disk Operating System (DOS), wildcards are used to replace one or many words. Wildcards can be identified by the following symbols * (star) and _ (underscore). Both wild cards provide the same purpose which is to replace one or more words. There is only one difference between the two wildcards which is the order in which the matching algorithm tries to match them with the words they replace (Wallace 2005). This will be discussed further in Section 4.3.2.

```
<aiml>
<category>
        <pattern>
                _ PRINTER DOES NOT * PRINT *
        </pattern>
        <template>
                Are you sure your printer is on?
        </template>
</category>
</aiml>
```

Figure 4.3 AIML Format with Wildcards

```
<category>
        <pattern>
        What are you
        </pattern>
        <template>
                I am a <bot name="order"/>
                I was activated at <bot name="birthplace"/>,
                on <bot name="birthday"/>.
                My <bot name="botmaster"/> was <bot name="master"/>.
                He taught me to sing a song.
                Would you like me to sing it for you?
        </template>
</category>
```

Figure 4.4 AIML Predicates

AIML also contain variables that can be set and retrieved at runtime known as predicates. The predicate values are stored in an XML file (Figure 4.5). Predicates can be utilised to set conversational agent properties such as the name, gender and birth place. Figure 4.4 highlights how predicates are utilised in AIML. Predicates such as the <that> tag can be utilised in order to maintain some sort of context in the conversation. This tag, however, only maintains the previous response the conversational agent provided to the user and does not take the whole conversation into context.

```
<?xml version="1.0" encoding="utf-8" ?>
<root>
        <item name="order" value="conversational agent" />
        <item name="birthplace" value="Department of Computer Science &
Information Systems, NMMU, Port Elizabeth" />
        <item name="master" value="Gopal Ravi Sankar" />
</root>
```

Figure 4.5 Predicate Values

Another tag, called SRAI, can be utilised to remap the input (Figure 4.6). The tag basically allows the execution of the input processor on what SRAI is given and uses that as output.

```
<category>
        <pattern>
                My printer is not working
        </pattern>
        <template>
                Are you sure there is power? Please check the power socket and the main
        power switch. If this information does not help, inform me.
        </template>
        <pattern>
                _ printer * not working
        </pattern>
        <template>
                <srai>
                        My printer is not working
                </srai>
        </template>
</category>
```

Figure 4.6 The <SRAI> Tag

AIML is stored in a data structure known as a graphmaster. When the conversational agent is loaded the AIML files are automatically loaded into this structure. The graphmaster plays a key role in the pattern matching process. The graphmaster pattern matching works similar to that of a person reading a dictionary, where the first letters of a word are used to find the word. If an exact match is not found then the system looks for wildcards that could satisfy the condition. This process is discussed in more detail in the next section.

4.3.2 ALICE Pattern Matching Algorithm

Most conversation agents utilise pattern matching to function (Lee, Sung and Cho 2001). Pattern matching was discussed in respect to speech in Section 4.2.1.1. This section however, will discuss this technique as it is utilised in ALICE (Wallace 2001, 2005).

AIML comes down to two basic subjects: what happens on the pattern side and what happens on the template side. On the pattern side the ALICE processes the users input and makes a decision as to what template to trigger.

When the user inputs text into the textbox and presses the enter key on the keyboard, pre-processing steps start occurring within the ALICE. The first process that occurs is known as deperiodisation. This process removes ambiguous punctuation marks from the sentence that was input by the user (Wallace 2001). Abbreviations such as Dr. and Mr. have the full stop punctuation marks removed from them. Heuristics are also utilised to split up sentences that are long.

Once sentences are split up, they are passed onto the next step which is known as normalization. In this process all words are put in upper case upon which contractions are expanded. Contractions such as “He’ll” and “She’d” will be expanded to “HE WILL” and “SHE WOULD”. Normalisation also ensures that there is exactly one blank space next to each word in the sentence and replaces iconography like “☺” with words such as “SMILE”. The final step of normalisation is to correct some basic spelling mistakes. The completely deperiodised and normalised sentence is now passed down to the AIML matching algorithm.

The AIML matching algorithm is responsible for searching through thousands of AIML categories in the chatbots brain (the graphmaster) to find the pattern that has the best match. The graphmaster contains a collection of nodes known as nodemappers. Nodemappers map the branches of each node. The branches can be either single words or wildcards. The root of the graphmaster can contain thousands of branches as each one represents the first word of all patterns in AIML. Figure 4.5 highlights a spiral of the graphmaster with 24000 branches of the ALICE AIML. The number of leaf nodes is equal to the number of categories and each leaf node contains a template tag. There are three steps that the matching algorithm utilises to match an input to a pattern. The inputs to the matching algorithm are:

1. An input string with word “X”
2. A Nodemapper of the graph.

The following steps are then utilised to match the input words in a string (Wallace 2005):

1. Does the Nodemapper contain the key “_”? If it does then the subgraph that is rooted at the child node linked by “_” must be searched. All remaining suffixes of the input following “X” must now be searched to see if one matches. If no match is found then try:
2. Does the Nodemapper contain the key “X”? If it does then the subgraph rooted at the child node linked by “X”, must be searched using the tail of the input. If there is not match then try:

3. Does the Nodemapper contain the key "*"? If it does, search the subgraph rooted at the child node linked by *. All remaining suffixes of the input following "X" must be searched to see if one matches. If a match is not found then traverse back up the graph to the parent of this node and put "X" back on the input of the head.

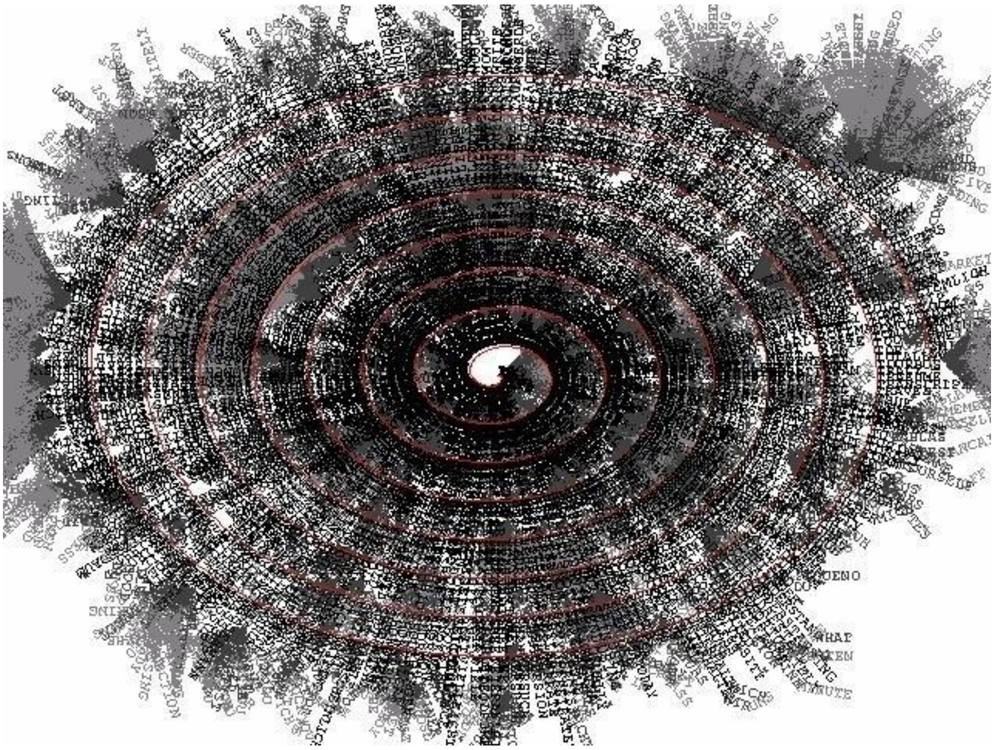

Figure 4.7 Branches of ALICE Brain with 24000 Categories Loaded.

In order for this pattern matching algorithm to be complete a terminal case is implemented where if the input is null (no more words) and the Nodemapper contains the <template> key, then a match is found. The search must be halted and the matching node must be returned. If the root Nodemapper contains a key "*" and it points to a leaf node, then the algorithm is guaranteed to find a match (Wallace 2005). The following must also be noted (Wallace 2001):

1. at every node the "_" has first priority. Exact word match has second priority followed by "*" which has the lowest priority.
2. the patterns in AIML need to be ordered alphabetically. In addition they must be ordered so that "_" comes before any word and "*" after any word.
3. the matching algorithm matches word-by-word and not category by category.
4. the AIML matching algorithm is a restricted version of a depth-first search known as back tracking.

Let us examine a few examples in order to understand the pattern matching algorithm:

If the Nodemapper contains the following three patterns:

1. _ IS A CAR
2. WHAT IS A CAR
3. WHAT IS A *

If the input is “what is a car”, then the first pattern will match because the underscore has priority over any specific word. On the other hand if the input was “what is a dog” then the third pattern will match as none of the other patterns contain the word “dog”.

If the Nodemapper contains the following three patterns:

1. _ IS *
2. WHAT IS A DOG
3. WHAT IS *

If the input is “what is a dog”, then the first pattern would match as the underscore wildcard takes first priority and will absorb the word “WHAT” and the second wildcard star will match the sequence of words “A DOG”. Even though the exact pattern exists, the underscore has the highest priority and therefore will override the exact match.

4.3.3 Shortcomings of ALICE and AIML

The biggest flaw of AIML is that it is very difficult to keep the context of a conversation with the user (Wilcox 2008). This makes ALICE very difficult to be utilised as an effective solution for automation problems faced at contact centres. Because of this problem the conversation often is “all over the place” (Wilcox 2008). The user may, for example, be taking about the weather at a particular point in time and ALICE will digress to talk about sports. This does not make it an effective solution to be utilised.

Ambiguity is still a problem in ALICE as it utilises a natural language for input. Users may have multiple ways in order to address their problems and if the AIML does not cater for all of these cases, the system would not recognise the input (Wilcox 2008).

Pattern matching in ALICE utilising AIML makes use of exact word matching. The process of normalisation only caters for very basic spell checking and therefore will not be able to cater for every single spelling mistake. If a list of synonyms could be afforded for certain words pattern matching may be much easier.

ALICE requires a very large knowledge base (AIML categories) in order for the system to keep the user involved in a conversation (Wilcox 2008). This brings about a problem of knowledge base compilation. Compilation of a knowledge-base can be done utilising a variety of tools online such as the Gaibot AIML editor (http://www.alicebot.org/downloads/tools.html). These tools, however, do not allow for parsing a text file of conversation which outputs an AIML file. Moreover, they do not cater for automatically placing wild cards in the relevant places in order to create default categories.

4.3.4 Benefits of ALICE and AIML

The benefits of ALICE and AIML will be discussed with respect to contact centres because this research investigates a natural language interface in particular a conversation agent as an automation tool for contact centres. The main advantage of ALICE as a conversational agent is that natural language can be utilised to state queries. This gives a natural interaction technique and a natural feeling as though one was interacting with another human being. Moreover it would be a better interaction technique than remembering menu options and keying in options over a phone.

Conversational agents provide a good solution to the cost versus satisfaction trade off as discussed in Section 2.5.3. Customers can engage in automated dialogue with conversational agents to effectively complete their tasks at much lower cost than human-assisted support (Lester *et al*. 2004). The obvious problem, however, is that conversational agents cannot operate in a variety of domains as there are limitations of natural language technologies. They can, however, operate in circumscribed domains while providing a cost-effective solution where question-answering requirements are bounded (Lester *et al*. 2004).

4.4 Applications of Natural Language Interfaces

NLIs have been utilised in a variety of domains. Section 4.4.1 discusses the applications that use conversational speech interfaces which is followed by an overview of text based conversational agents (Section 4.4.2).

4.4.1 Conversation Speech Interface Applications

Speech recognition technologies have matured to an extent that various applications have been able to take advantage of it (deWet *et al.* 2003). The following subsections discuss some of the applications of speech recognition and synthesis via a conversational speech interface.

- Contact centres - conversational speech interfaces have been utilised in an ad hoc fashion in order to replace IVR. IVR systems are usually proprietary software and therefore speech is just added as an extra component to the current IVR structure (Larson 2005). This, therefore, does not alleviate any of the problems associated with IVR, but only changes the input modality. Where contact centres completely overhaul their IVR to support conversation speech interfaces utilising domain specific language (a subset of natural language) users often struggle to utilise these systems as they would not understand what acceptable input for the systems is.
- Mobile phones - mobile phones have had conversation speech interfaces for the past decade or so. These interfaces have been utilised as voice commands for features such as voice dialling and call answering (Shaikh and Chitode 2003). When utilising voice dial$$ing the user trains a voice tag for single entry in the phone book. The phone then automatically places the call if it identifies the tag. Call answering is utilised to either accept or reject a phone call. When the user's mobile phone rings the user can utter a keyword which accepts the phone call or declines it. Small screens and keys on mobile phones also pose problems for users to utilise these devices efficiently (Zhou *et al.* 2004). A conversational speech interface would allow for user to complete tasks such as sending a short message service (SMS) or an email a lot quicker and more efficiently and effectively.
- Applications for the disabled - people who have disabilities can find that conversational speech interfaces to be useful. If people have no means by which they can type out messages, they can use speech recognition to voice commands to their mobile phones or personal computers (Schroeder 2004).

4.4.2 Text Based Conversational Agents

The advances in computational linguistics have allowed for well-engineered conversational agents to play an increasing role in the automation of certain business processes. Conversational agents are now capable of having a descent conversation with users in assisting them to accomplish their tasks (Lester *et al.* 2004). Clear communication is critical for acquiring, serving and retaining customers. Conversational agents provide clear communication with users and provide a good customer service/cost relationship for enterprises. Enterprises are employing conversational agents as cost effective solutions as they are able to engage in simultaneous conversations with users while providing good customer service (Lester *et al.* 2004). These conversational agents are used in various domains and were used by the author to find flaws:

- Hanover Conversational Agent - was created by Daden Limited, which assists clients with real estate advice (Chatbot List 2007). It utilises text based input and provides text and speech as output. This chatbot had more than 10,000 hits on the first week of launch and the company claims that it has been very successful and has drastically reduced costs associated with their contact centre. The chatbot is very interactive but seems to lack basic intelligence about the domain. Two questions were posed to the bot (Figure 4.8) which were related to the domain but were not answered correctly (providing the right information).

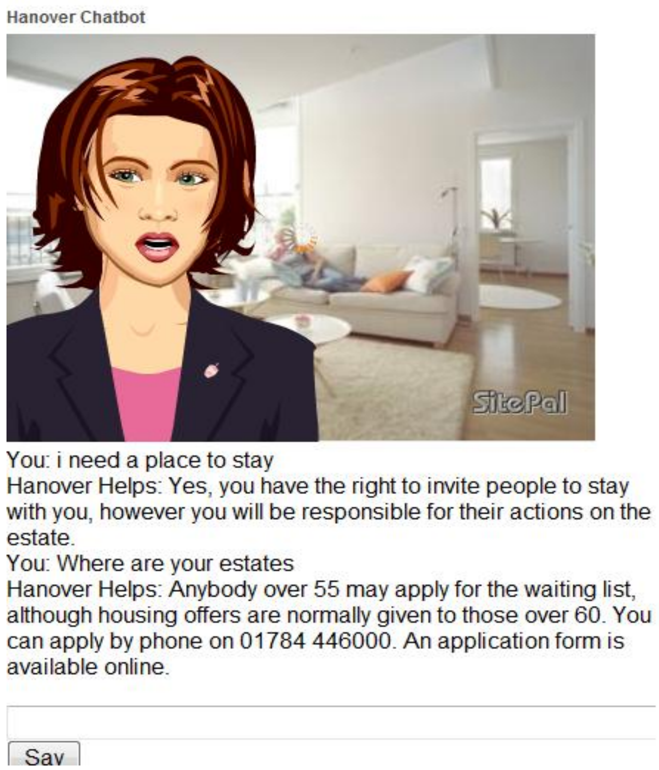

Figure 4.8 The Hanover Chatbot

- Best Western and Virgin Holidays conversational agents - The Best Western Hotel group (Figure 4.9) and Virgin (Figure 4.10) utilise agents developed by Synthetix (Chatbot List 2007). They required a solution that increased customer service while reducing costs (Chatbot List 2007). Both companies are located in the travel and leisure industries and therefore are listed together. Both agents have a wealth of information about hotels, leisure facilities and even nearby airports.

  The one problem with these agents is how they respond to questions that they are not sure about. They list a series of questions which the user has to scan through and select the appropriate one (Figure 4.9). Experts suggest that this is not natural language interaction but rather direct manipulation (Ogden *et al.* 1996). If the system does not understand a query it must rather ask questions that will derive the information from the user. Another problem with these conversational agents is the fact that you have to provide your first name before proceeding with any queries. Users sometimes would not like to provide personal information especially over the internet.

- Jeeney AI - Jeeney is a conversational agent that was programmed to carefully dissect the English language in order to understand what is being said to her. However upon conversing with this conversational agent it is quite obvious that she would not be ideal for completing user tasks as conversations drift and does not stick to one topic. Moreover the chat bot does not utilise spell checker and if the user makes spelling errors the user input will not be recognised. Figure 4.11 highlights the Jeeney systems interface.

Figure 4.9 Best Western Chatbot

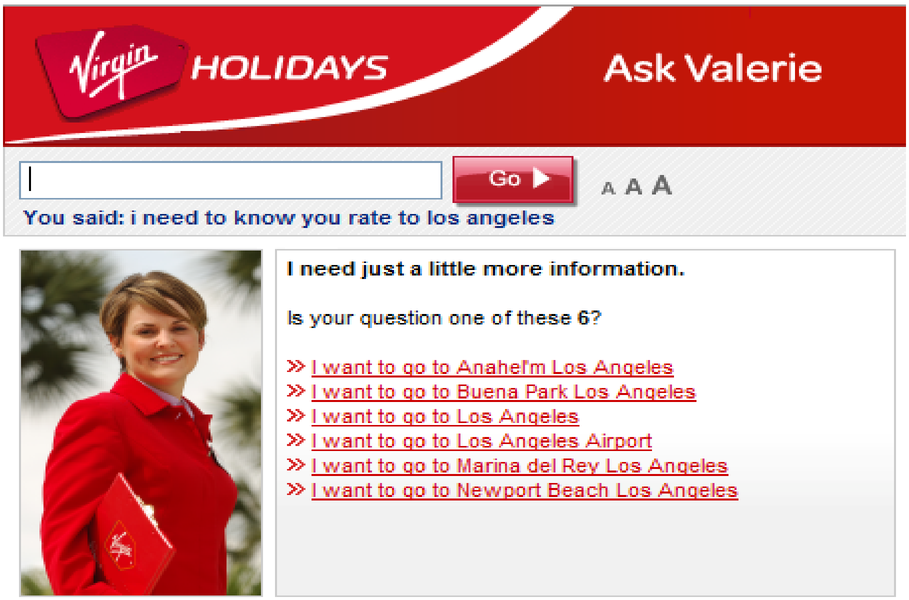

Figure 4.10 Virgin Holidays Chatbot

Figure 4.11 Jeeney AI

It is still clear that existing conversational agents, although very novel, still have various limitations which need to be addressed. The shortcomings of NLI , as discussed earlier in the chapter, are still prevalent in these implementations.

4.5 Conclusions

The goal of this chapter was to highlight the benefits and shortcomings of natural language interfaces. This was highlighted by introducing two types of NLIs and their various advantages and disadvantages. The main disadvantage of these interfaces was discussed as being the ambiguous nature of natural language while the main advantage was that users can utilise everyday language to interact with the system.

The advances made in natural language processing, have allowed for natural language interfaces to play a prominent role in enterprises. These companies are always striving to strike a balance between cost savings and good customer interaction. Natural language interfaces allow for this as normal agents would not have to be utilised as virtual conversational agents are able to do the job cheaper and almost as effectively. Companies are generally moving towards text based interfaces as speech recognition technology is still maturing (Lester *et al*. 2004).

Conversational agents have to satisfy two main conditions if they are to be effective in an organisation: they must provide good output to any user input and they must integrate well with the organisations business processes.

Chapters 3 and 4 gave an overview on rule-based expert systems and natural language interfaces, identifying their strengths and weaknesses. Interactive voice response systems and conversational agents make use of these concepts in attempts to implement automated processes in contact centres.

Chapter 5 will discuss the process that was followed, which led to the development of an NLI and an RBS, which was followed by the proposal of two Hybrid models (which incorporated RBS and NLI mechanisms). One of the models were implemented and evaluated (Chapter 6).

Chapter 5 - Design and Implementation

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later -- Scott Rosenborg

5.1 Introduction

Contact centres are the first point of contact with a company by a client, after the purchase of a product or service (Chapter 2). Customer service is therefore a critical success factor at these centres. High call densities and personnel costs have required contact centres to turn to automation techniques in order to service repetitive and simple queries by clients. Automation has also been employed to get rid of long queues which forces clients to wait for a contact centre agent to service their calls. IVR is an example of an automation technique utilised to service clients. IVR resembles and shares the characteristics of a forward chaining RBS (Chapter 3). It has a variety of advantages and disadvantages as discussed in Sections 2.4 and 3.4. The main advantage of IVR is that it has a long term memory of user actions because of the tree structure being utilised (Figure 2.4). The major disadvantage of IVR is that users are restricted to menu options. The consequent poor quality of interaction with customers often results in a low resolution of queries.

NLIs can be utilised to alleviate the problems brought upon by IVRs through the flattening of the menu structure (Chapter 4). This would be possible as users would make use of natural language to address the system and this would rid the automation system of menus. The use of natural language however brings about a problem in ambiguity (Fum *et al.* 1988). The various methods that could be utilised to combat ambiguity were discussed in Section 4.2.3.

Conversational agents which are a type of an NLI could be used for the purpose of automation at contact centres and would result in cost savings and reductions. ALICE is an example of a conversational agent and will be used as the case study in this research. The benefits and shortcomings of the ALICE conversational agent were discussed in detail in Section 4.3.3. The main benefit of this conversational agent is that users can utilise natural language to interact with it. The major shortcomings were highlighted as ambiguity and short term memory. Table 5.1 compares the advantages and disadvantages of IVR and NLI side by side.

Natural Language Interface (ALICE)	Rule Based Expert System (IVR)
✓ Users can use natural language to state their Queries.	✗ Users input are restricted to menu options.
✗ Input can be ambiguous as user input is unlimited.	✓ Ambiguity is not a problem as user input is restricted.
✗ System has a short term memory and poses a problem for users.	✓ System has a long term memory.
✓ Users input not limited by menu choices	✗ Users have to remember menu options.
✓ NLIs allow for a more natural interaction with the system.	✗ System does not feel natural as input is restricted.
✓ **Advantage**	✗ **Disadvantage**

Table 5.1 Advantages and Disadvantages of NLI and RBS

This chapter first sets out to report on the process of confirming the advantages and disadvantages of RBS and NLI systems. This was done through the implementation and evaluation of two prototypes (Section 5.2). Consequently a Hybrid approach is introduced by presenting and comparing two different models (Sections 5.3 and 5.4). Finally the implementation of one of these models is discussed in Section 5.5.

5.2 Confirmation of Advantages and Disadvantages of NLI and IVR [1]

In order to confirm the theory regarding advantages and disadvantages of NLI and IVR (Table 5.1) two prototypes were implemented. Before the two prototypes could be implemented (Section 5.2.2) data had to be gathered in order to compile the knowledge needed for both prototypes (Section 5.2.1). Finally the evaluation process was conducted (Section 5.2.3).

5.2.1 Knowledge Base Compilation

ALICE requires an AIML knowledge base (Section 4.3.1) and RBS require the knowledge of an expert to be transferred to a system utilising inference rules in order to function. There was therefore a need to conduct a data gathering session.

Initially participants had to be selected in order to gather knowledge to set up the AIML knowledge base for the ALICE system. Participants were not necessary to gather data for the IVR system as knowledge needed to be gathered from expert sources and transferred to the system. The sample for the data gathering process was random and convenient in nature so that it would truly simulate a heterogeneous demographic diverse group of customers that deal with contact centres on a daily basis. A total of fifteen participants were chosen for the data gathering process: six undergraduate students, five administrative personnel at the Department of Computer Science and Information Systems at NMMU and four postgraduate students. Nine participants were female while six were male. Figures 5.1 and 5.2 represent these statistics.

The data collection process consisted of collecting data for the NLI prototype as well as the IVR prototype. The establishment of the ***knowledge base for the NLI*** prototype (ALICE) involved setting up case studies related to the research domain of printer problems: no paper in the printer, paper jam and toner problems. See Appendix A for the case studies.

---

[1] This work on the advantages and disadvantages were published in the conference proceedings of the SATNAC conference. (RAVI SANKAR, G., GREYLING, J. and VOGTS, D. (2008): *Towards a Conversational Agent for Contact Centres*. *South African Telecommunications, Networks and Applications Conference (SATNAC)*. BARNARD, D., BARNARD, L., DEVILLIERS, P. and GOOD, R. Wild Coast, Eastern Cape, South Africa, Telkom. **1**.)

[2] This work on the Hybrid models and their comparisons was published in the conference proceedings of the SAICSIT

Data was collected by setting up a dummy contact on Google Talk called NMMU Help Desk. Participants chatted to this contact by working through the case studies presented to them. The person representing the help desk contact was the author of this dissertation. The Google Talk instant messenger was utilised as it allowed for chats to be stored for later use (Figure 5.3). Participants were encouraged to interact with this contact without any hindrance (they could state their queries in any manner they wished).

The data collected was then converted to AIML using a parser. The parser was written in C# .Net. The inclusion of AIML wildcards was not catered for. This was done manually through the use of a text editor.

Data for the IVR prototype was collected through personal communications with various technical personnel at the NMMU help desk and at the Department of Computer Science and Information Systems. A variety of printer manuals were also consulted in order to understand how troubleshooting of printer problems was described. This information was then consolidated and utilised in the implementation of the prototype.

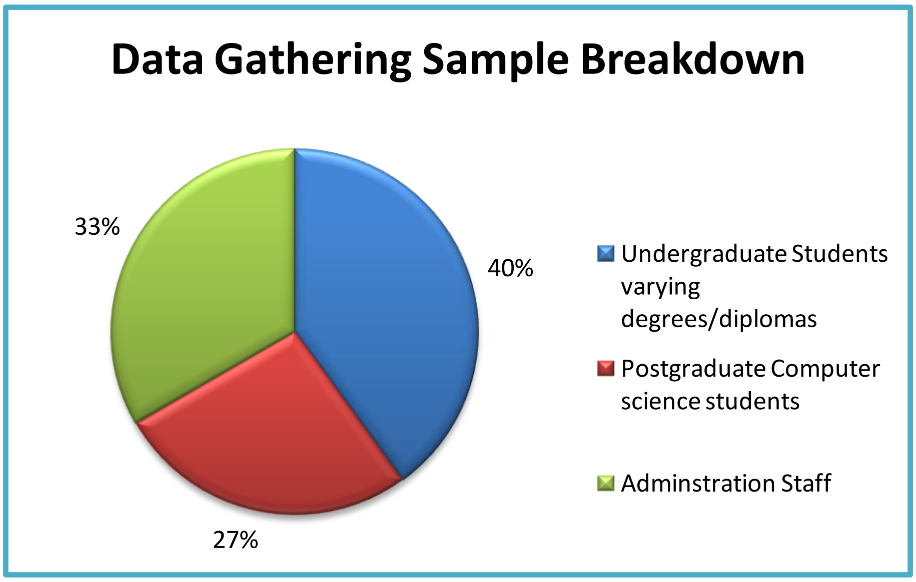

Figure 5.1 Breakdown of Data Gathering Sample in Accordance to Occupation (n=15)

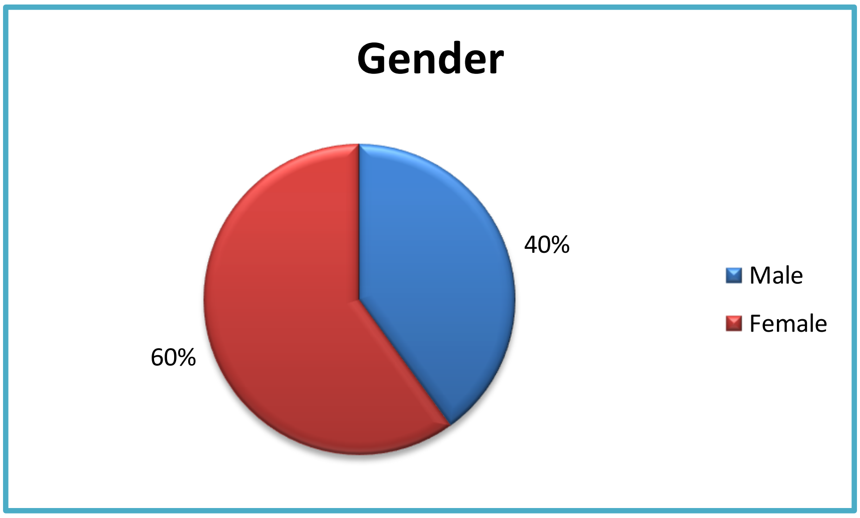

Figure 5.2 Gender Profile for Data Gathering Sample (n=15)

bassoncj: hi i was busy printing a document, and about 20 pages in it just stopped all together and now has a light flashing

what can i do?

me: Are u sure you have paper in your printer

11:49 AM **bassoncj**: im not 100% sure, how would i check?

me: Look for the Labels on the front of the printer e.g 1 & 2 can u see it

11:50 AM **bassoncj**: ok i see them, but what am i supposed to be doing with them?

me: I need you to look to the left of 2 and pull out the tray gently. can you do that?

11:51 AM **bassoncj**: the tray?

Figure 5.3 Chat History with a Data Gathering Participant

5.2.2 Implementation of Prototypes

Two prototypes were developed representing NLI and IVR. This will be addressed in the next two subsections.

5.2.2.1 ALICE (NLI)

Implementation of the knowledge base for ALICE system was done through the utilisation of a parser that converted the chat histories into AIML. Upon the completion of the AIML sets, an interpreter for AIML was acquired through the GNU Public Licence. A user interface was created utilising the dynamic link library (DLL) file, which resembled an instant messaging interface (Figure 5.4).

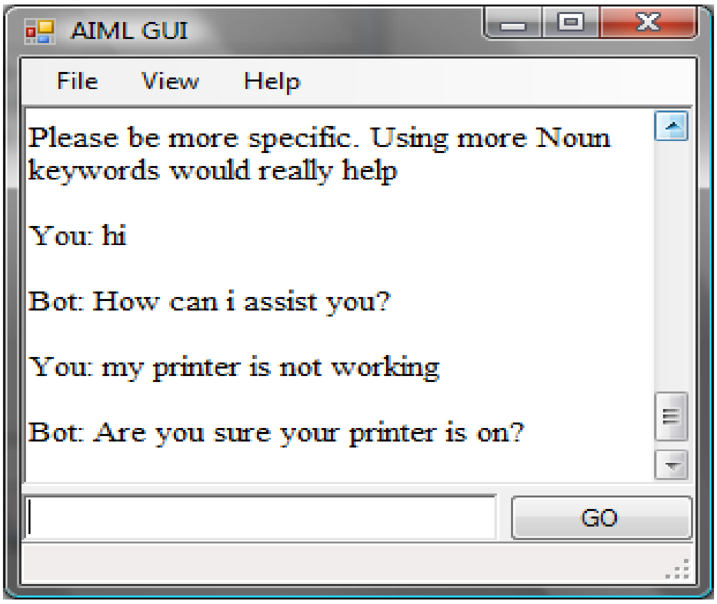

Figure 5.4 ALICE System utilising an IM Interface

The system was then tested by the author against the AIML sets to determine whether the system ran smoothly. A logger was implemented to analyse conversations. The logger was used to log all conversations between a user and the prototype. If the user had a conversation with the conversational agent and did not receive an output, this could be addressed by analysing the log.

5.2.2.2 IVR (RBS)

The implementation of the RBS was done utilising a forward chaining expert system shell known as CLIPS. The prototype was modelled to resemble an IVR. The data that was collected was converted into rules that the CLIPS inference engine could understand. A typical rule is depicted in Figure 5.5. A screenshot of the interface of this system is depicted in Figure 5.6. The prototype was then tested to ensure that every menu option was functioning correctly and the right paths were taken in accordance to user input.

```
(defrule begin
 (declare (salience -9000))
 ?x <- (start) =>
 (printout t crlf crlf crlf
    "The program is designed to help in diagnosing computer hardware problems" crlf
    "The program will take you step by step through the procedures to troubleshoot the"
crlf " specific problems with your hardware." crlf crlf
    "Please state specific hardware you have problems with." crlf
    "  1) Printer" crlf
    "  2) Monitor" crlf
    "  3) or exit the program" crlf
    "Choose 1 - 3 -> ")
 (retract ?x)
 (assert (hardware type =(read)))
 (assert(ck-select)))
```

Figure 5.5 A Rule in CLIPS

5.2.3 Evaluation of Prototypes

A summative study was conducted in order to compare the NLI with the RBS in order to deduce the advantages and disadvantages of utilising these different concepts. A summative study involves evaluating a system or prototype to understand how well it meets its functionality objectives and *comparing systems or prototypes to one another* (Tullis and Albert 2008).

```
The program is designed to help in diagnosing computer hardware problems
The program will take you step by step through the procedures to troubleshoot the
 specific problems with your hardware.

Please state specific hardware you have problems with.
  1) Printer
  2) Monitor
  3) or exit the program
Choose 1 - 3 -> 1


Does the printer have any electrical power? y or n -> y


Is there paper in the printer? y, n or dn(dont't know) -> dn


Who is the manufacturer of your printer

 1) Canon
 2) Samsung
 3) HP

Choice ->
```

Figure 5.6 RBS Utilising an IVR Approach

The goal of this evaluation was to ascertain the advantages and disadvantages of utilising an NLI and an IVR for contact centre automation and compare it those discussed in Table 5.1.

5.2.3.1 Evaluation Methods

The metrics that were to be evaluated were efficiency, effectiveness and satisfaction of the user in order to measure the usability of the prototypes. These metrics were chosen as customer service is critical and therefore the prototypes have to be able to complete the task by supporting user goals while providing a satisfying experience.

Different mechanisms have been utilised to evaluate conversational agents, ranging from black box evaluation that evaluates a system as a whole and glass box testing which evaluates the individual components of a system. These mechanisms, however, do not measure user satisfaction, efficiency or effectiveness but rather that of the architecture of the system and therefore cannot be applied to our evaluation.

The Loebner prize is the most well-known test for conversational agents and utilises the Turing test to evaluate chatbots. The test evaluates the ability of the conversational agent to fool users into believing that they are talking to a human. Judges are allocated ten to fifteen minutes with a conversational agent and will then evaluate it in terms of naturalness. The Loebner prize got its idea from Alan Turing's paper "Can Machines Think?" (Turing 1950). Turing proposed an imitation game in which a human observer tries to guess the gender of two players, one of which is a man and the other a woman, but will be screened such that the observer would not be able to tell by voice or appearance. Turing goes on to suggest that if a machine replaces one of the two players and the observer is not able to distinguish between human and machine, this can be taken as strong evidence that the machine can think.

Sceptics, however, believe that this is not an effective test for chatbots as it relies solely on fooling people. Some sceptics believe that intelligence cannot be determined by surface behaviour (Sheiber 1994). Moreover, the Loebner prize is restrictive in terms of topics that judges could utilise and this was also frowned upon.

Munteneau and Boldea utilised the following six questions to evaluate dialogue systems (Munteanu and Boldea 2000) and these were also utilised in this evaluation. These questions were used as they evaluate the metrics the author wanted to measure in this usability evaluation. The questions asked to the participants of the evaluation are highlighted in Table 5.2.

	Question	Effectiveness	Efficiency	Satisfaction
1	Did you get all the information you wanted using the system?	✓		
2	Do you think the system understood what you asked?	✓		
3	How easy was it to obtain the information you wanted?		✓	
4	How easy was it to reformulate your questions when you were invited to?		✓	
5	Do you think you would use the system again?			✓

Table 5.2 Questions Used to Evaluate Usability Metrics

Once the evaluation metrics were established and questions for the post-test questionnaire were in place, the next step involved selecting the participants for the evaluation.

5.2.3.2 Selection of Participants

The only criteria utilised was that the users had to be different from those utilised in the data gathering stage as they supplied the data and would find the systems easier to use. Sixteen participants were selected and they were heterogeneous in nature. This sample was split as follows: four administration staff, four postgraduate students and eight undergraduate students. The sample contained eight males and eight females. These stats are depicted in Figures 5.7 and 5.8.

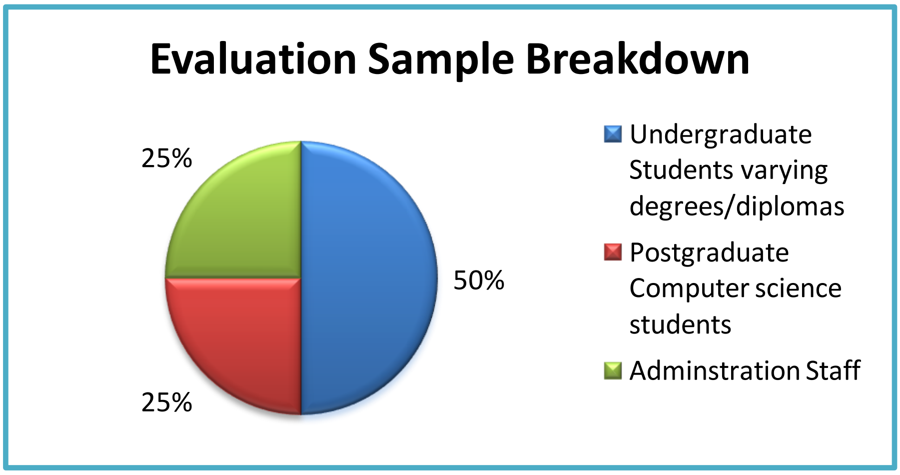

Figure 5.7 Breakdown of Evaluation Sample in Accordance to Occupation (n=16)

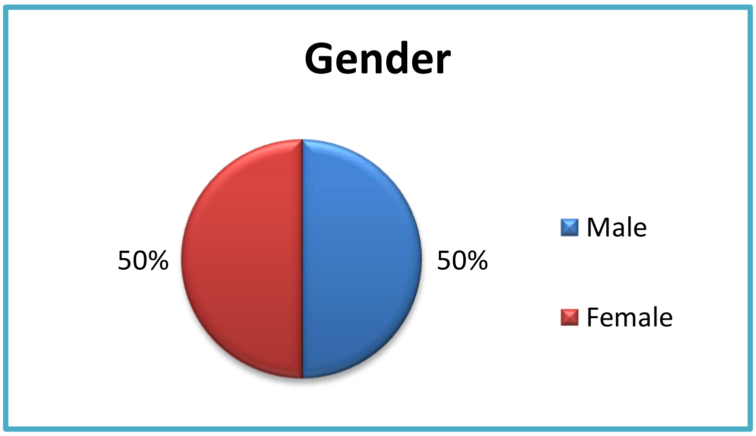

Figure 5.8 Gender Profile for Evaluation Sample (n =16)

5.2.3.3 Evaluation Instruments and Procedure

User testing was done on the participant's computer in their offices. Participants interacted with the NLI first and then interaction occurred in the form of menus to test the IVR.

The instruments used in the evaluation included case studies and a post-test questionnaire. Users would act upon the case studies and fill out the post-test questionnaire after completing the case studies on both prototypes. The same case studies were used on both prototypes and were the same ones used in the data gathering phase (Appendix A).

5.2.3.4 Results of Evaluation

Once all participants were tested, the questionnaires were analysed using thematic analysis as the evaluation was qualitative in nature. The main observation from the participants testing the NLI is their frustration at having to rephrase some of the answers they had given the conversational agent while it was troubleshooting their problems. As can be seen from Table 5.3, 31% of the participants had trouble with this. This is mainly due to fact that a large AIML set is needed to conduct the study successfully. The participants also identified that the conversational agent sometimes changed conversation completely and did not solve the problem. This was because ALICE has a short memory span and does not remember the past conversations held with a user. The problems noted above by the test participants were all mentioned in Table 5.1.

Complaint	Percentage
Had to rephrase questions	31%
Conversation did not make sense	25%
Did not remember my previous answers	13%

Table 5.3 Top 3 Problems Encountered when Utilising NLI (Ravi Sankar, Greyling and Vogts 2008a)

The main observation made from the participants testing the IVR prototype was that it took them far too long to solve their problems. Some of their options they had to select were the last ones and they felt it unnecessary to view all the other options. The main reason users did not like this system (Table 5.4) was because they found it time consuming (56% of the users mentioned this). Participants also stated that they felt that too much information was directed at them which made them feel lost at times.

Complaint	Percentage
Time consuming	56%
Too much information	50%
Long frustrating menus	31%

Table 5.4 Top 3 Problems Encountered when Utilising IVR (Ravi Sankar *et al.* 2008a)

It can be observed that slightly more participants felt that the IVR prototype made it easier for them to find information (88% vs. 82%) (Table 5.5). NLIs though, were easier to use to obtain information and were also rated higher as systems which users would use again. 69% of the participants were more satisfied with the NLI in comparison to the RBS.

IVR is identified as being restrictive in terms of input whereas NLI is viewed as being less restrictive since it simulates a natural dialogue. Since an NLI system allows users to use natural language to convey their queries it could save time and money since queries would be resolved in a shorter space of time.

	NLI	IVR
Users were able to find information	82%	88%
Was it easy to obtain information	90%	80%
Would you use the system again?	82%	50%
Overall Satisfaction	69%	31%

Table 5.5 Comparison of NLI and IVR (Ravi Sankar *et al.* 2008a)

It is evident that the results of the evaluation process confirmed the advantages and disadvantages of IVR and NLI, as concluded from the literature survey. Furthermore, it is noted that the advantages of IVR potentially overcome the disadvantages of NLI and vice versa. Therefore a Hybrid system that combines the two different concepts should theoretically result in an improved automation system. Two models that could be used in implementing such a Hybrid system are discussed in the next section.

5.3 Hybrid Prototype Models [2]

The main disadvantage of an NLI (ALICE) is that it has a short-term memory while IVRs have a longer term memory as every input is considered before producing an output. However IVRs are not very user friendly as input is usually restrictive whereas NLIs are user friendly allowing users to utilise natural everyday language when interacting with the system. When discussing a Hybrid approach, there are two types of models that could be used: An NLI using RBS concepts and an RBS using an NLI.

5.3.1 Natural Language Interface with Rule-Based Expert System Concepts

The problems faced by ALICE as an NLI was discussed above as being short-term memory. This problem is brought upon by the simple pattern matching that the ALICE conversational agent uses.

A pattern (input) is matched to a certain template (output); therefore no previous conversation with the conversational agent is taken into consideration. AIML predicates such as the <that> tag (Section 4.3.1) are used to somewhat maintain context, however are only able to store the most recent pattern in memory. This poses a big challenge when trying to solve problems that need the context of a conversation to remain static.

ALICE loads its knowledge base into a structure known as the graphmaster (Section 4.3.2). The graphmaster can be viewed as a large knowledge source where the root node can be viewed as a starting point for every single input (Figure 5.9). Moreover, if the same pattern is encountered users would be given the *same* output (Figure 5.9). Figure 5.10 depicts this with the aid of AIML. If output is randomised to the same input, the order of the conversation would not make sense. A solution for this would be the utilisation of the graph structure that IVR utilises in order to solve problems for the user. If this structure can be

[2] This work on the Hybrid models and their comparisons was published in the conference proceedings of the SAICSIT conference. RAVI SANKAR, G., GREYLING, J., VOGTS, D. and DU PLESSIS, M.C. (2008): *Models towards a Hybrid Conversational Agent for Contact Centres*. *South African Institute of Computer Scientists and Information Technologists (SAICSIT)*. CILLIERS, C., BARNARD, L. and BOTHA, R. Wilderness Beach Hotel, Wilderness, George, Western Cape, South Africa, ACM. **1**:200-210.

implemented into an NLI it would be possible to maintain the context of the conversation (Figure 5.11).

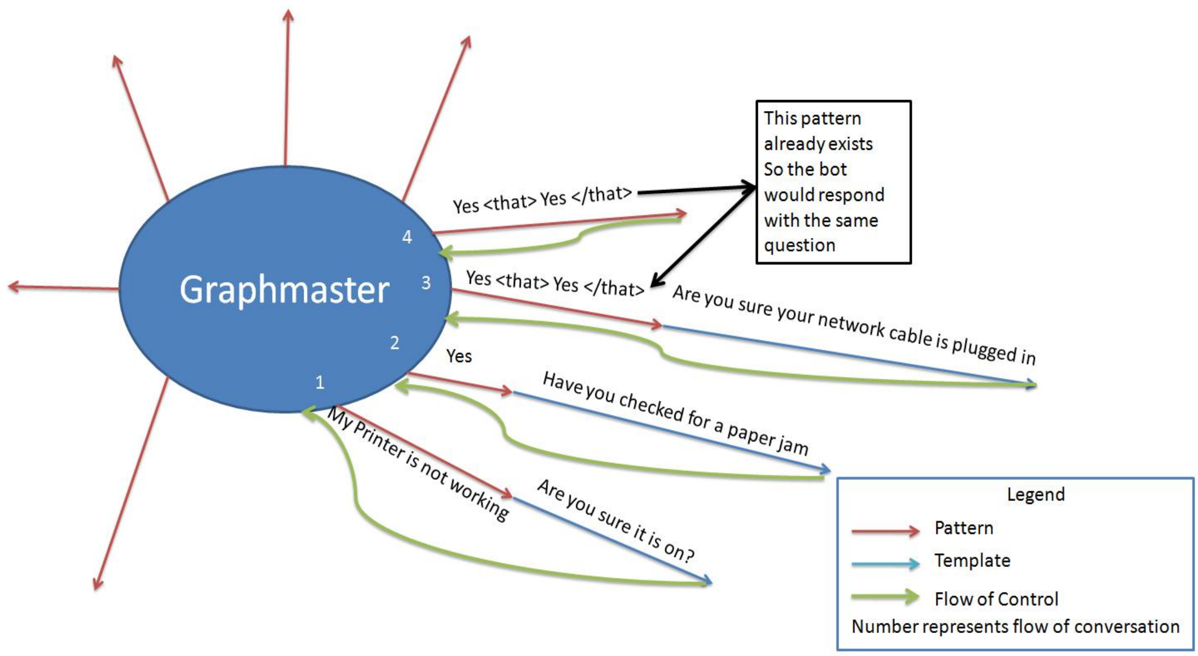

Figure 5.9 Simplified Graph Structure of ALICE

The limitation of a short-term memory can be addressed through the utilisation of a rule base and a natural language interface such as ALICE. Using a rule base an NLI which utilises RBS concepts would allow users to use natural language to convey their commands and would not lose the context of the conversation. The NLI will utilise ALICE concepts of pattern matching with the only difference being that patterns are post tagged at runtime with the letter/word combination. This post tag will describe the *current state and context* of the conversation with the user. Rules would consist of a combination of numbers and letters. These numbers and letters describe the *branch and child of a tree* that the conversation has traversed to at a particular point in the conversation.

The approach described keeps the same graph structure of ALICE and this is seen as an advantage as someone with very little programming experience can utilise this technique to setup a knowledge base for the ALICE conversational agent that would be able to converse with a user while maintaining the context of the conversation.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<aiml>
        <category>
                <pattern>
                        My printer is not working
                </pattern>
                <template>
                        Are you sure it is on
                </template>
        </category>
        <category>
                <pattern>
                        Yes
                </pattern>
                <template>
                        <that> My printer is not working</that>
                        Have you checked for a paper jam?
                </template>
        </category>
        <category>
                <pattern>
                        Yes
                </pattern>
                <template>
                        <that>
                                Yes
                        </that>
                        Are you sure the network cable is plugged in
                </template>
        </category>
        <category>
                <pattern>
                        Yes
                </pattern>
                <template>
                        <that>
                                Yes
                        </that>
                        We will dispatch a technician
                </template>
        </category>
</aiml>
```

Same pattern, so chatbot would not know what to do.

Figure 5.10 Conflicts with the Same Pattern in AIML

The use of rules would make similar patterns seem different to the AIML interpreter as rules are tagged to patterns so no two patterns would look alike (Figure 5.11). This approach will also enhance the memory structure of ALICE while keeping its original pattern matching structure.

Post tags are depicted graphically in Figure 5.11 and through the use of AIML in Figure 5.12. If the ALICE system asks a user "***Are you sure your printer is on?***" the user has three things that he/she can say either: ***yes, no or don't know***. These three responses can be viewed as three different branches of the tree namely *A*, *B* and *C*. According to the choice made, further questions can be asked which can also be viewed as children of the parent branches and further post tagged. The tree structure is similar to that utilised by IVR systems which give them context and long term memory.

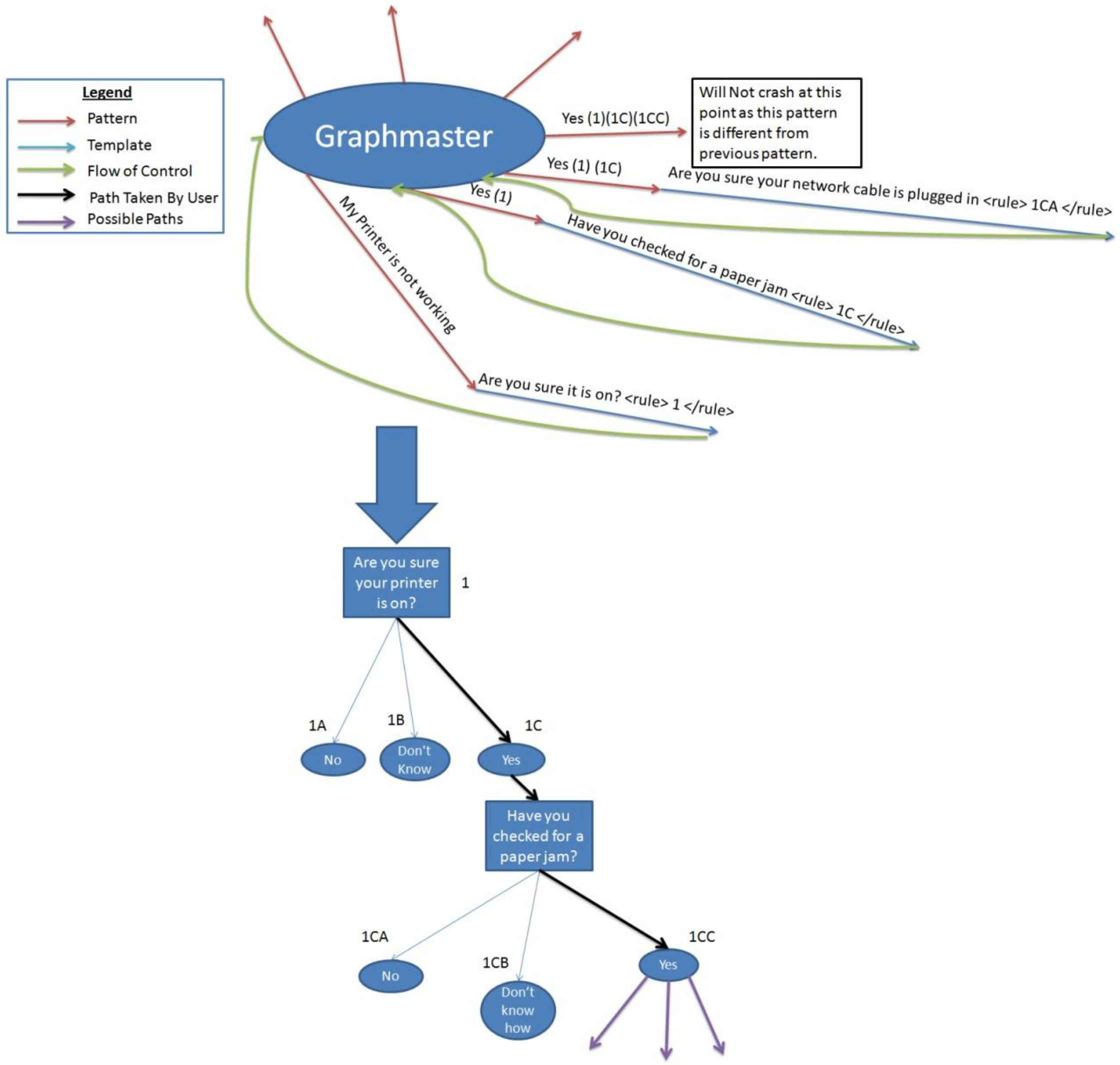

Figure 5.11 Simplified Graph Structure with Rules Compared to IVR

```
<aiml>
<category>
        <pattern>
                MY PRINTER DOES NOT WANT TO PRINT MY
DOCUMENT
        </pattern>
        <template>
                Are you sure your printer is on?
                <rule>1</rule>
        </template>
</category>
<category>
        <pattern>
                YES (1)
        </pattern>
        <template>
                Please locate tray 2 and once located please pull the tray out
                by locating the grip underneath the tray and pull it out. is there
                any paper in the tray?
                <rule>1C</rule>
        </template>
</category>
<category>
        <pattern>
                YES (1)(1C)
        </pattern>
        <template>
                Please check your control panel. On the two line display, does
it indicate a paper jam
                <rule>1CB</rule>
        </template>
</category>
</aiml>
```

Same Pattern will look different as rules are post tagged to it

Figure 5.12 AIML with Rules

5.3.1.1 A Model for a Natural Language Interface with Rule-Based Concepts

The model shown in Figure 5.13 involves two main components which are the *user interface* and the *RBS architecture*. *The RBS architecture* is broken down into three components: ***knowledge base, working memory and the inference engine*** and the *user interface* displays input by user and output by the system. It must be noted that the RBS architecture discussed is similar to that discussed in Section 3.3.

The ***knowledge base*** consists of AIML files which are used by the inference engine (ALICE client). As discussed in Section 4.3.1 AIML stores the patterns and templates to facilitate the conversation between the user and the system. The AIML will only cater for printer problems and specifically only the following problems: No Paper, Paper Jam, Printer Off and Toner Problems. The ***working memory*** consists of rule files which store all current rules utilised by the system. Furthermore the rules will be utilised in the post tagging process of users input. The ***inference engine*** utilises ALICE pattern matching in order to produce output to user input (provided in natural language). The ALICE pattern matching process was discussed in Section 4.3.2. This component has three main functions (Ravi Sankar, Greyling, Vogts and du Plessis 2008b):

- it tries to locate rules that could be utilised to guide conversation;
- allows for it to maintain context in the conversation; and
- it checks the path taken by the user and shows how the conclusion was reached and communicates this to the users by displaying a result on the graphical user interface.

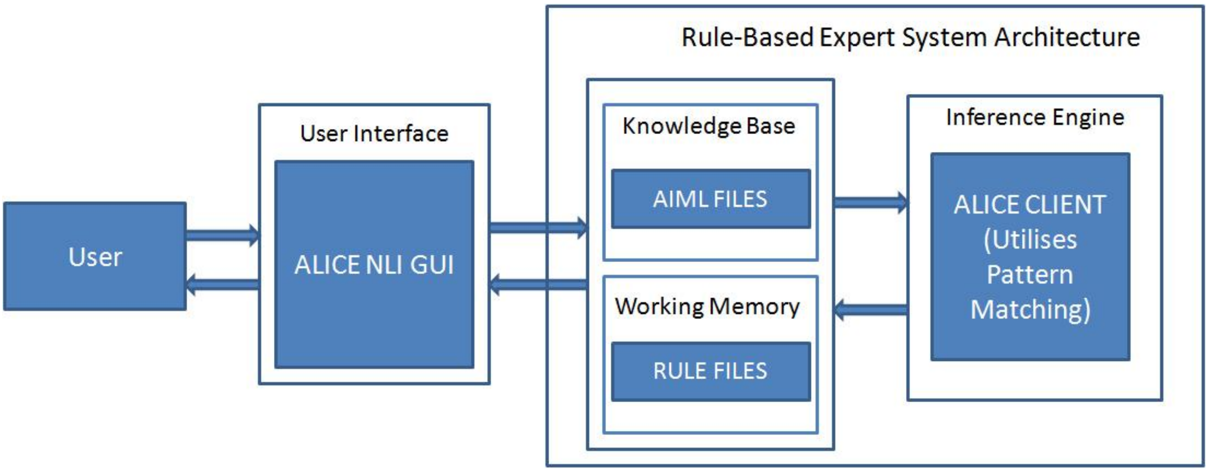

Figure 5.13 Rule Based Natural Language Interface Model (Ravi Sankar *et al.* 2008b)

5.3.2 Rule-Based Expert System using a Natural Language Interface

IVRs which are an example of a forward chaining RBS gather information through the use of menu options which are selected by the user through either numbers or short phrases (Giarrantano *et al.* 1989). The functioning of an IVR can be compared to that of a tree structure. Every question an IVR poses is answered using a menu option. Once a menu option is chosen, the next node in the tree is chosen according to the user input. Information is therefore gathered in a *particular order*. Figure 5.14 depicts the functionality of an IVR system with respect to its tree structure.

The use of a NLI will allow users to volunteer information without the use of a menu interface. The overall time the user takes to accomplish a task would be shorter as the user would have answered fewer questions posted by the system (Abraham 2005). This is because the user can volunteer more information than is necessary whenever the expert system presents a question to the user.

This system would make use of a natural language module that would use a parser which would be responsible for interpreting user statements as facts. The natural language module will utilise ALICE pattern matching techniques to provide the RBS module with facts. These facts will then act as input for the rule based expert system module. If more facts are provided than necessary then additional facts will be stored for later use. These additional facts will be analysed by a controller module (which will be found within the RBS module), so as to make sure that questions regarding the facts supplied are ignored as they have already been provided. If there are no useful facts that could be generated from user input, the natural language model will engage in some "fact-finding" questions so as extract facts from the user. Once the facts are gathered and fed to the rule based expert system, it triggers a particular rule which is sent to the natural language module. This goes back to the AIML to trigger the relevant template that is needed. Figure 5.15 depicts the process that occurs in the natural language module.

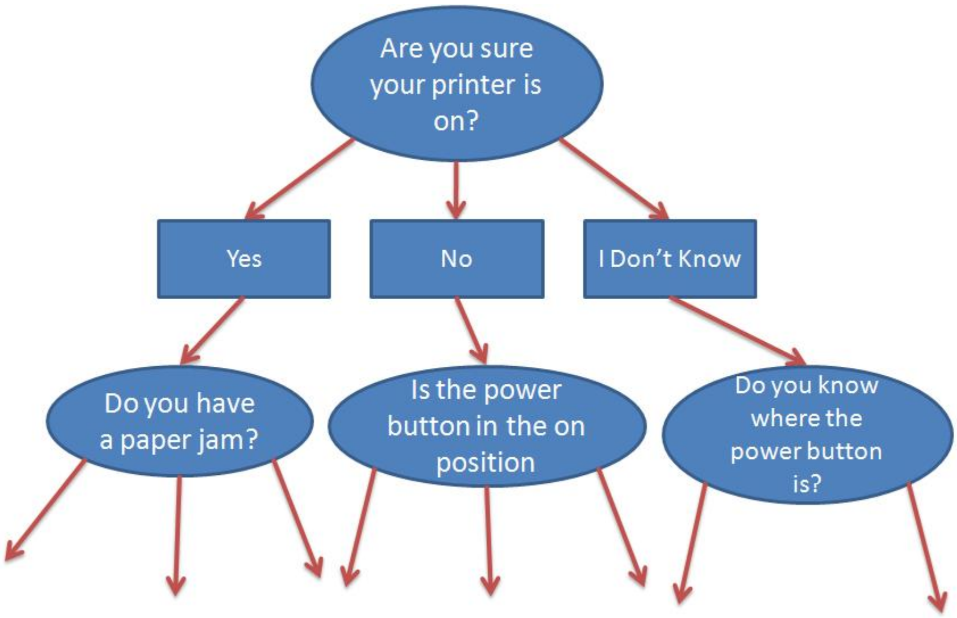

Figure 5.14 Graph Structure of an IVR

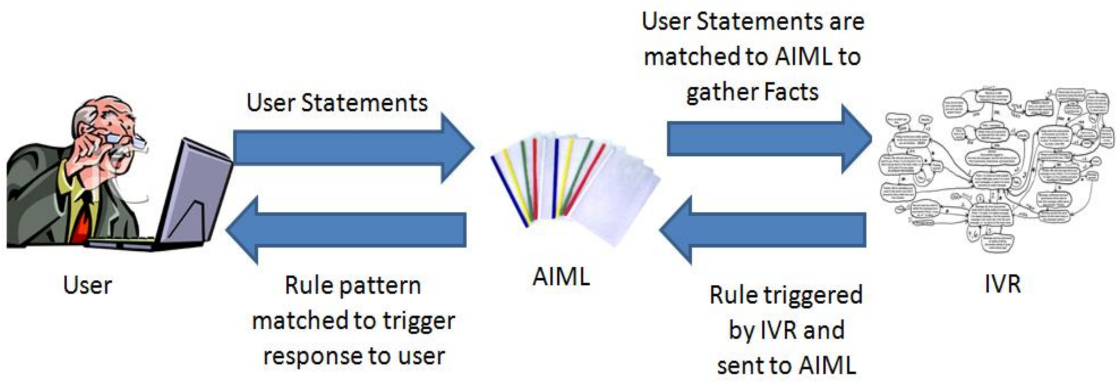

Figure 5.15 Inner Working of the Rule Based Expert System using a Natural Language Interface

An NLI for RBS uses this same principle. Each node can be envisioned as a mini ALICE that tries to gather relevant information from the user and then traverses down to the next mini ALICE it has to utilise to solve a task. Each mini ALICE would only store AIML sets related to the questions it would need to ask, therefore increasing the efficiency of the system.

Unlike the structure used by ALICE (Figure 5.9), Figure 5.16 highlights the new graph structure utilising this new approach. This structure allows for the user inputs to be taken into consideration (i.e. stored and used as facts) before proceeding to the next node of the graph to further diagnose a problem. If the user volunteers more information than necessary it will be stored and analysed before asking the next question. This would be necessary as the next question the system may ask may not be necessary, as the user may have already volunteered the information. Therefore this system would not lose context and would also have a long term memory which is currently lacking in the ALICE system. It also rids the system of menu options which the user would have to remember through the use of everyday natural language.

5.3.2.1 Model for a Rule-Based Expert System using a Natural Language Interface

The model depicted in Figure 5.17 has two main components which are the natural language interface module and the expert system module which utilises a rule base. The NLI will utilise pattern matching concepts to provide responses to user input. The expert system will utilise user inputs to be converted into facts for the expert system to be able to manipulate user information to output rules. The rules are then sent to the NLI module which will use pattern matching to produce a result for the user.

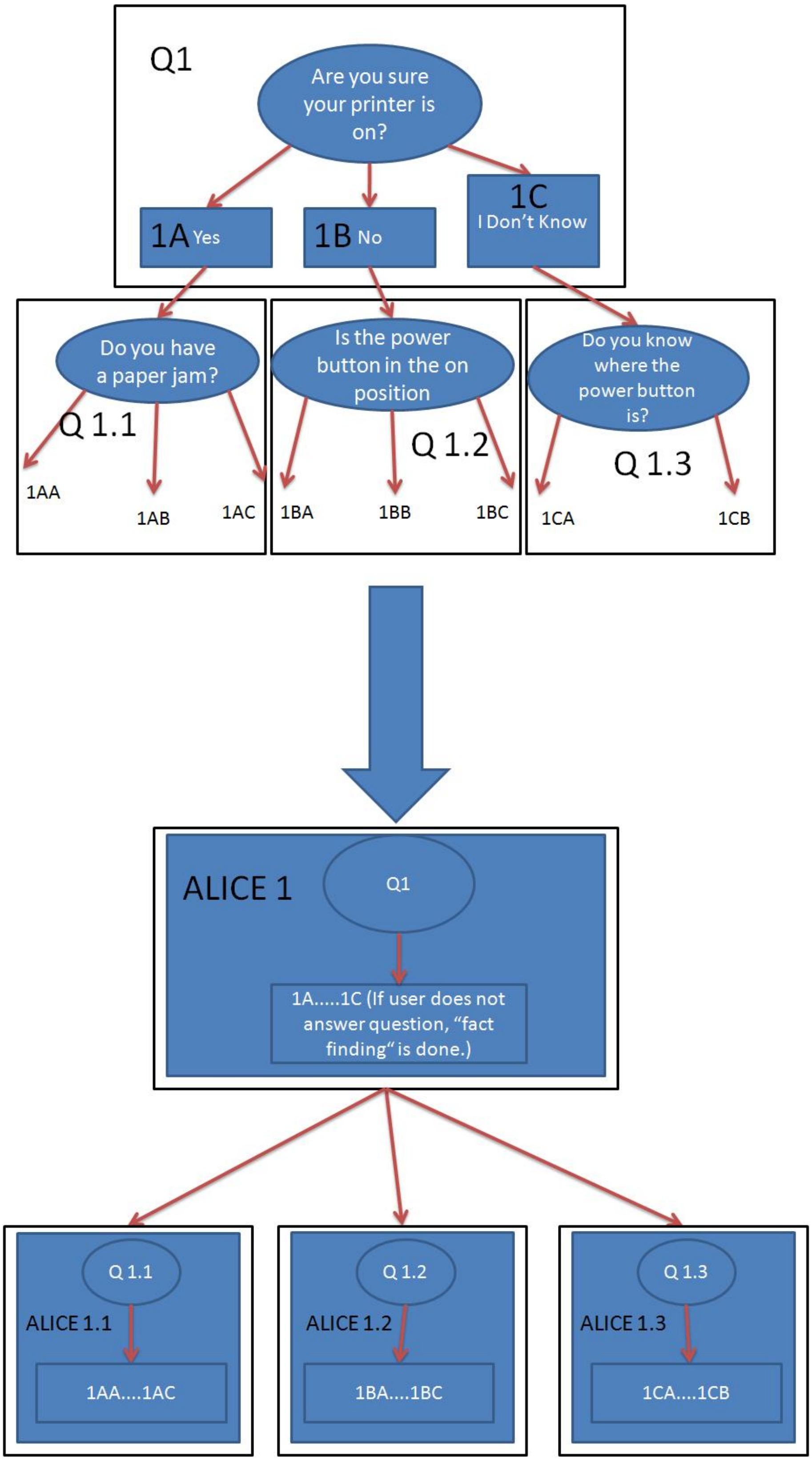

Figure 5.16 Graph Structure of an NLI for RBS

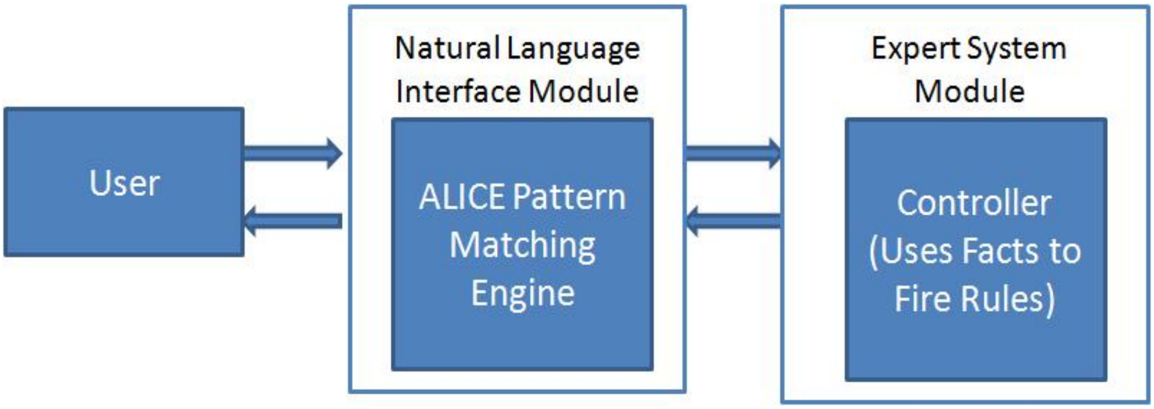

Figure 5.17 Rule Based Expert System using a Natural Language Interface Model (Ravi Sankar *et al*. 2008b)

The various components will now be discussed in further detail (Ravi Sankar *et al.* 2008b):

- ***Expert System***- this module takes input from the natural language module and the sub-module controller. Facts are drawn from the NLI module to produce rules which are taken in as input in the module.
    - *The Controller* - this sub-module is responsible for storing additional facts that the user volunteers. If the user volunteers more information than necessary, facts provided in user input will be stored. Thereafter each following question will be referenced to analyse whether the next step is necessary or not. If the next step involves acquiring information from the user that has already been given, it will be repetitive in nature. When there is a lack of information (facts), clarification dialogues will be put forward to the user.
- ***Natural Language Interface*** - this module is responsible for utilising ALICE concepts of pattern matching to allow users to use natural language to interact with the rule-based system. It sends its input to the rule-based expert system which converts user input into facts. The facts are used to draw rules used by this module to produce an output for the user. This is achieved through pattern matching. AIML which contain all possible rules that the RBS can output will be contained as patterns with relevant outputs.

Two Hybrid models were presented. A comparison of these models will now follow.

5.4 Model Comparison

The two models presented are different, as one model is a natural language interface while the other a rule based system. However they were both presented to accomplish the same task. Table 5.6 highlight the important comparisons between the two models.

Natural Language Interface using Rule-Based Expert System Concepts	Rule-Based Expert System Using a Natural Language Interface
NLI would have a large, unstructured knowledge base.	Knowledge base will have structure as rule based concepts will be utilised.
Permutations and combinations involved in creation of rules for AIML.	Rules will be coded in the rule based system, so no need for rules in AIML.
Will not be efficient as knowledge based would be large.	More efficient as AIML for only a subset of conversations will be stored in each ALICE.
Utilises the same graphmaster and rules are easy to learn and create.	Graphmaster differs from original. More than one graphmaster utilised therefore memory intensive.

Table 5.6 Comparison of Hybrid Models

The model for NLI using RBS concepts (Section 5.3.1) is implemented using concepts unique to ALICE. This, however, brings about the problem of having a large, unstructured knowledge base. The rules give the knowledge base some structure, but the creation of the knowledge base would involve the creation of the AIML sets with the inclusion of rules. The structure of a user's conversation can never be determined. Therefore there would be various permutations and combinations of these rules that would have to be taken into account due to the way in which keywords may be picked up. This approach may also not be seen as being efficient as the knowledge base would be quite large and could slow down the response of the system. The knowledge base for this system would be the same as that of ALICE and would therefore be easy to create if one understood the domain well.

The model for RBS using an NLI (Section 5.3.2) is implemented using concepts which are derived from expert systems. These systems are however very difficult to use and generally take a long time to complete a task. The introduction of a natural language can significantly reduce the time taken to accomplish a task. The controller module gives the system context and would allow the system to store previous conversations conducted with the user. The ALICE chatbot is only used for its pattern matching capabilities and therefore the knowledge base would not be as large as the first model. There would be no need for the various permutations and combinations of rules as the system is a rule based expert system. Efficiency of this system would also be better than the previous model, due to smaller sizes of knowledge bases. The system would however need to make use of two sets of AIML (user statements to facts and rules to statements) and this could be complex and tedious to create and very tedious as rules and facts have to map exactly to ones stated in AIML.

The first model was picked for implementation (an NLI using RBS concepts). The reasons for the choice are that it would directly answer the primary research question as the second model is in essence an RBS, while the first is an NLI (which is the solution we wish to achieve). The first model is also a more practical and cost saving option for contact centres as anyone who understands the domain would be able to implement the AIML with little training. The ALICE graphmaster structure will also remain the same (Figure 5.11) therefore only AIML has to be changed, unlike the second model where changes have to be made to both systems and two sets of AIML have to be present. These AIML sets will also be difficult to set up as every fact and rule have to be present in the correct syntax without which the system will not function properly.

Section 5.5 discusses the implementation of the Hybrid prototype where NLI makes use of RBS concepts.

5.5 Implementation of the Hybrid Prototype

This section highlights the various processes that were followed in the implementation of the Hybrid prototype. The process consisted of four stages, namely establishing the structure for AIML, implementing AIML to maintain context, ascertaining implementation issues and conducting a pilot study.

5.5.1 Establishing Structure for AIML

The main inputs for the system as depicted in Figure 5.13 are the AIML and Rule files. The AIML sets are tagged with rules in order to maintain context in the conversation. An inspection of the menu structure which the IVR prototype uses was conducted to model the rules for the Hybrid system. The rules used by the Hybrid system therefore follow the menu structure found in IVR. Figure 5.18 highlights how the menu structure of the IVR was utilised to create the rules. Rules were established at every user decision point, which was every time the user had to make a choice for his/her input. For example, if the user had a printer that was not working, but it was on, and the user had not checked for a printer jam, according to Figure 5.18 the current rules that would be appended would be (1)(1A)(1AB). This not only tells the system the steps that have already been solved. Furthermore, the log files will record the exact steps the user had taken in order to solve the problem.

5.5.2 Implementing AIML and Maintaining Context

The AIML structure utilised in the purely pattern matching system was used to map it on to a tree as described in Figure 5.18. The logs which stored the conversations from the evaluation (Chapter 5) were utilised to update the AIML for cases where output could not be found. The AIML was then tagged with the rules utilising a custom AIML predicate called <Rule> (Figures 5.11 and 5.12). This <Rule> Tag tells the system that it is going to traverse a certain new branch in the tree. The ALICE DLL that was acquired through a GNU public licence allowed for easy creation of custom tags and is shown in Appendix F.

Custom tags enhance the AIML vocabulary to include more functionality (Wallace 2005; Tollervey 2006). Functionality can be programmed by the user and can be used to manipulate strings found in the pattern. Custom tags are found within the <template> tag of AIML. Figure 5.19 example highlights how custom tags are utilised in the Hybrid prototype.

The <rule> predicate will use the information inside its tag to post tag the rule to the next pattern. These rules will then give the conversations structure and will be able to solve the problem in a structured manner.

The next step was to alter the normalization process (Chapter 4) that ALICE uses to rid the user input of all punctuation marks. This had to be done as the rules that were post tagged were found in brackets and this would not get past the normalization process. The ALICE interpreter which is implemented in C#. Net uses regular expression to rid the user input of unwanted charaters and the regular expression was modified to allow the following brackets;“(and “)” to pass through the normalization without being stripped. The Normalization code can be found in Appendix F.

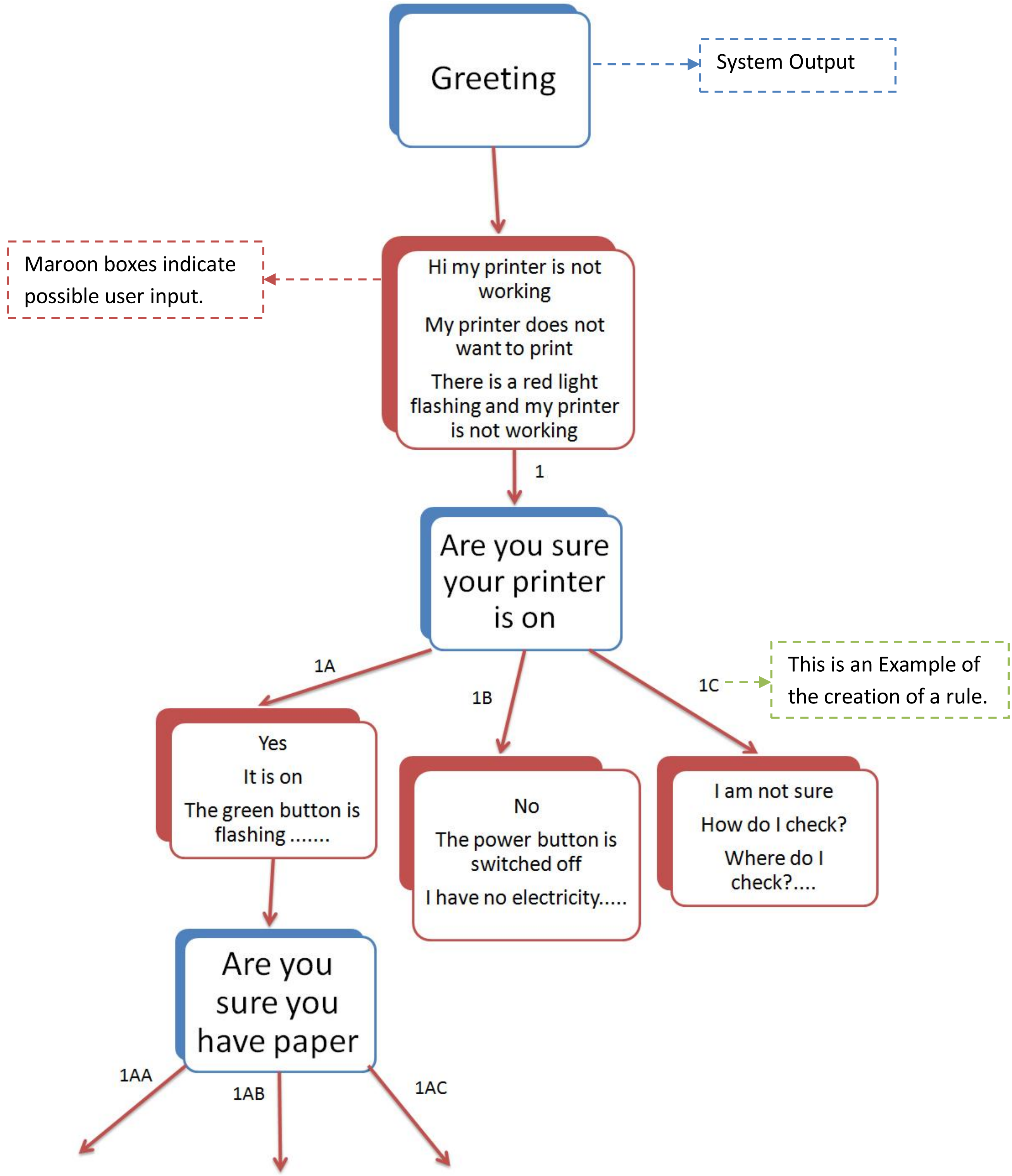

Figure 5.18 Menu Structure Utilised to Create Rules

```
<category>
        <pattern>
                MY PRINTER STOPPED PRINTING MY DOCUMENTS
        </pattern>
        <template>
                Are you sure your printer is on?
                <rule>1</rule>
        </template>
</category>

<category>
        <pattern>
                HOW DO I CHECK IF IT IS ON (1)
        </pattern>
        <template>
                Please locate the power button on your printer. Once
located see if it is in the "on" position. Is it in the "On" position?
                <rule>1A</rule>
        </template>
</category>
```

Figure 5.19 Custom Tags as used in the Hybrid Prototype

5.5.3 Implementation Issues

The main issues with implementation were the various permutations and combinations of the AIML that had to be constructed. It is very difficult to determine every possible input of the user. Therefore the system had to cater for instances where input was not recognised and guide the user to provide input that the system would recognise. In most cases when this occurred, the system would ask questions that required just a simple *yes/no*, after which the system functioned normally. When the system was tested by the implementer there were no signs of slowing due to the highly unstructured knowledge base. The next section will discuss the pilot study conducted in order to determine whether the system functioned properly and without errors.

5.5.4 Pilot Study

A pilot study was conducted utilising experts to test the Hybrid prototype for errors. Experts were chosen as they would be able to test the prototype for errors thoroughly as they know the steps to be taken when fixing a printer. The process followed will be explained in the next few subsections.

5.5.4.1 Goals

The goal of the pilot evaluation was to test the Hybrid system for obvious faults such as system crashes, unexpected input handling and lack of clarity when explaining solutions. This was achieved through participants acting on case studies presented to them.

5.5.4.2 Selection of participants

Participants were selected to resemble experts who have the domain knowledge of being able to fix printer problems themselves rather than calling on the help of experts. Experts were carefully chosen and were classified as those who had utilised computers, printers and instant messaging for over 4 years. They needed to have at least 3 years of printer troubleshooting experience as well. The sample was convenient in nature and consisted of postgraduate computer science students from the Department of Computer Science and Information Systems at NMMU. Six participants were chosen of which five were male and one was female.

5.5.4.3 Development of Tasks

The task list can be found in Appendix B and consisted of the following three printer solving scenarios: printer off, no paper and paper jam. Efficiency was not tested through time. Efficiency was tested by noting the extent to which the prototype supported participant's goals (being able to complete a task).

5.5.4.4 The test procedure

Six scripts that explained the case studies were prepared beforehand. The purpose of this was to ensure that each script given to participants were the same and contained the same instructions and information. Participants were briefed on the goal of the study and were invited to explore the Hybrid system before starting their tasks. Once done the purpose of the post test questionnaire was explained. It was also explained that the test was qualitative in nature and that more than a simple yes/no answer would be needed.

Data was collected in the form of interaction logs and post test questionnaires. The post test questionnaires were qualitative in nature. The logs collected interaction data such as beginning and end time of a particular task and actual conversations between the participant and the prototype. The latter was done in order to edit the AIML for the final evaluation process. Upon completion of the various case studies users were asked to fill in the questionnaire and ask general questions regarding the prototype.

5.5.4.5 Results

The results will be addressed question by question and the various remarks will be listed. The questions posted to the participants were the same as those utilised in the previous evaluation of the NLI and IVR prototypes.

Did you get all the information you wanted using the system?

- *"Yes, there was enough information to assist in solving the problems"*
- *"Yes, It was easy to resolve problems quickly"*
- *"Yes, however some bugs prevented me from getting all the information"*
- *"First task completed correctly with no issues. The other test cases encountered problems as some my responses were not recognised."*

It can be noted that from the above remarks, that the participants were able to get most of the information that they were seeking. The prototype however did have a few bugs as one of the participants stated that their input was not recognised.

Do you think the system understood what you asked?

- *"Not all the time..."*
- *"Well, I think it did, however I had to rephrase my responses sometimes"*
- *"Yes, it responded quickly"*
- *"Yes responses were relevant and to the point"*

The participants generally seem to be in a spilt opinion as to whether the system understood them or not. This is highlighted in the remarks that have been highlighted above.

How easy was it to obtain the information you wanted?

- *"Fairly simple"*
- *"Was able to get most information"*
- *"Sometimes the system did not understand what I wanted"*

The information that participants were seeking was quite easy to obtain. Some participants, however, did state that the prototype did not provide all the information they wanted as it was not able to understand the participants.

How easy was it to reformulate your questions when you are invited to?

- *"I did not reformulate anything"*
- *"No. It was easy to reformulate"*
- *"Very easy to reformulate"*

Participants were able to reformulate queries to the prototype quite easily. Questions had to reformulated, when the prototype was not able to understand the input provided by the participant.

The fifth question asked participants if they would utilise the system again and all of them said they would while some went on to mention that they would definitely utilise it when all the bugs are ironed out. The last question addressed the overall satisfaction of the prototype and four out of the six participants said that they were satisfied with the prototype while the remainder stated that they would be happy when the bugs are fixed.

The positive aspects of the system as stated by the participants are:

- *"Easy interface to use"*
- *"When running correctly it seem that it can be a simple and efficient "self-service"/ DIY solution to rectifying problems"*
- *"Easy to use and system understands me well.*
- *"Because of the way it asks you questions, it appears as though you are chatting to a human being and there is therefore a level of trust established with the system"*

The negative aspects as listed by the participants included:

- *"The system does not seem to understand some of the input."*
- *"A few bugs here and there"*
- *"Input capabilities limited. I would like to be able to chat to system rather than typing"*

Generally it can be concluded that the Hybrid prototype had limited knowledge in terms of patterns. This was addressed by checking logs and catering for unknown input. The crashes referred to by the participants were caused by the lack of patterns for the ALICE bot to respond to. This was addressed by giving a default response "I need more information to respond to your query" so that users get constant feedback from the prototype. Once all the bugs were fixed and knowledge bases updated, the final evaluation was conducted. This process will be discussed in detail in Chapter 6.

5.6 Conclusions

The objectives of this chapter were to compare the advantages and disadvantages between IVR and NLI through the implementation and evaluation of prototypes that utilise these concepts, to introduce two models that could be used in the creation of a Hybrid system and to discuss the implementation of one of the models.

The implementation and evaluation of the two prototypes confirmed the advantages and disadvantages of IVR and NLI obtained from literature study. Two Hybrid models (an NLI using RBS concepts and an RBS using an NLI) were presented, as well as a comparison of the two models.

The NLI using RBS concepts model was selected for implementation. The main reason for the choice of model was that anyone who understands the domain would be able to implement the system with minimal training. Rules were implemented through the use of the <rule> custom tag and this gives the system structure and users would be able to solve their problems and understand the various steps they had undertaken.

A pilot study was conducted to test the Hybrid prototype to test for any errors. Data was gathered through the use of post-test questionnaires. The data was qualitative in nature and the various responses gathered were presented in Section 5.5.4. Overall reaction to the prototype was positive in nature and the various negative responses were utilised to improve the prototype.

The next chapter will address the final evaluation conducted in the research. The final evaluation compares the three prototypes (IVR, ALICE and Hybrid) to determine which would provide a better solution as an automated option at contact centres

Chapter 6 - Evaluation

And the users exclaimed with a laugh and a taunt: "It's just what we asked for but not what we want." -- Anonymous

6.1 Introduction

The previous chapter proposed two models that could be utilised in the implementation of a Hybrid prototype as an automation technique at contact centres. The implementation of one of these models was also presented and a pilot study was conducted to determine if the prototype ran smoothly. This chapter reports on the final evaluation of the Hybrid prototype with the IVR and NLI prototypes.

The Hybrid prototype utilises NLI concepts which are combined with RBS concepts. It is, however, essentially a conversational agent and needs to be evaluated as such. An investigation into evaluation methods and techniques were performed and discussed in Section 5.2.3.1. Similar questionnaires were administered in evaluating for the Hybrid prototype as with the IVR and NLI prototypes.

The goal of the evaluation was to answer the following questions (Rogers, Sharp and Preece 2007; Tullis *et al.* 2008):

1. Do the users find the Hybrid prototype more ***effective*** than the other prototypes? Does it ***support the goals*** they wish to satisfy?
2. Do the users find the Hybrid prototype more ***efficient*** than the IVR and NLI prototypes? How much ***effort*** does the user need to put in to complete a task?
3. Is the user more ***satisfied*** with the Hybrid prototype than with the IVR and NLI prototypes?

The evaluation aimed to collect both quantitative and qualitative data to measure the efficiency, effectiveness and satisfaction of the three systems and determine through statistical analysis and ranking data which is the preferred solution. The ultimate goal will be to answer the primary research question as stated in Section 1.3.1: *Can a natural language interface address the limitations and enhance the benefits brought about by IVR as an automation technique at contact centres?*

6.2 Statistical Instruments

A variety of statistical instruments was used to measure the usability metrics for the purposes of the final evaluation and is discussed below.

6.2.1 *T*-tests

A *t*-test is used to determine whether means from two samples are significantly different in order to conclude that both samples come from a single population (Dunn 2001; Tredoux and Durrheim 2002). This research used a *t*-test for dependent samples (repeated measures t-test). This test is utilised when independence between two datasets cannot be ensured (Tredoux *et al.* 2002). An example of a dependent sample would be a sample of 20 participants testing two different prototypes and time is measured to determine which prototype is more efficient. The two samples of time are from the *same* 20 participants and therefore, the samples are dependent and not independent.

6.2.2 Wilcoxon Matched Pairs test

The Wilcoxon Matched Pairs test is used to test whether two related samples have the same median. The test is non-parametric in nature in comparison to the parametric *T*-test. In a parametric test observations must be drawn from a normally distributed set of data (Dunn 2001). This could not be done due to the small sample size (n=20) for the final evaluation and therefore a non-parametric test was used where the population does not necessarily need to be normally distributed (Dunn 2001).

6.2.3 Analysis of Variance (ANOVA)

The *t*-test is used to measure differences between means of two samples. The Analysis of Variance (ANOVA) is used to measure for differences between the means of more than two samples (Tredoux *et al.* 2002). An example of ANOVA test is the comparison of the mean times to complete the same task on three different web designs in order to determine which one is the most usable.

6.2.4 Tukey Honestly Significant Difference (HSD) Test

The Tukey HSD test is used when the results from the ANOVA test have significant differences (Dunn 2001). When comparing means, the Tukey HSD test allows one to conduct pairwise comparisons between means. Pairwise refers to comparisons made between two means at a time (Tredoux *et al.* 2002). An example of a Tukey HSD test is comparing the mean time with the mean rating that a user gives when testing a prototype. If there is a significant result then it can be assumed that time impacted the score the participant gave in the test.

6.2.5 Spearman Rank Order Correlation

Correlation is a single number that describes the degree of relationship between two variables (Trochim 2006). An example when correlation can be used is to see if there is a relationship between time taken and rating of a prototype. In this instance one would want to determine whether there was a relationship between time taken to complete a task and the rating the participant of the test scored on a Likert scale. When ordinal data is used, such as Likert scales, correlation is measured using a Spearman Rank Order Correlation. Correlation is always between -1.0 and +1.0. A negative correlation means that there is a negative relationship while a positive one points towards a positive relationship.

6.3 User Testing

The aim of user testing was to assess three metrics namely effectiveness, efficiency and user satisfaction. The three metrics were measured in order to determine to what extent the prototypes supported user task completion.

The goal of user testing was accomplished using post-test questionnaires, ranking of the prototypes and timing of tasks. The post-test questionnaire contained questions shown in Table 6.1 which measured the three metrics of usability.

6.3.1 Usability Measurements

A post-test questionnaire was utilised to measure the effectiveness, efficiency and satisfaction by recording user reactions using a five-point Likert scale. Participants were presented with statements which were related to each of the three usability metrics (Section 6.2). They would then rate these statements on a five point Likert scale with the values (Rogers *et al.* 2007) as indicated in Table 6.1.

Six separate questionnaires were utilised with the order in which the different systems were tested differed. The post-test questionnaire consisted of three sections. Section A was utilised to gather the biographical data from the participant such as age, computer experience and printer troubleshooting experience. Section B measured the usability of the three prototypes with respect to effectiveness, efficiency and satisfaction and also asked participants to rank the prototypes from best (1) to worst (3). Section C was the qualitative part of the questionnaire which was used to gather general comments about the three prototypes.

	Question	Effectiveness	Efficiency	Satisfaction
1	*Did you get all the information you wanted using the system?* **[Never (1).....Always(5)]**	✓		
2	*Do you think the system understood what you asked?* **[Never (1).....Always(5)]**	✓		
3	*How easy was it to obtain the information you wanted?* **[Very Easy (1).....Very Difficult(5)]**		✓	
4	*How easy was it to reformulate your questions when you were invited to?* **[Very Easy (1).....Very Difficult(5)]**		✓	
5	*Do you think you would use the system again?* **[Definitely Yes (1).....Definitely No(5)]**			✓

Table 6.1 Statements used in Post-Test Questionnaire and their Relation to Usability

The Likert scale was reversed for some of the questions to prevent users selecting arbitrary values to finish the evaluation as quickly as possible. The sample questionnaire can be found in Appendix C. Table 6.1 highlights the questions asked in the evaluation.

6.3.2 Evaluation Instruments and Procedure

The evaluation instruments that were utilised in this evaluation were the case studies which participants would act upon to complete tasks and the post-test questionnaire which was used to gather qualitative and quantitative data from the participants.

The evaluation procedure was done in four stages: a briefing, task completion, post-test questionnaire completion and a debriefing. Each participant took about 35 minutes to complete the evaluation process. Testing was conducted at the user's office with the use of a personal laptop.

Participant's actions were captured using automated logs which contained the start and end times of each task and the actual conversations that took place between the user and the prototypes. This was done for the NLI and Hybrid prototypes in order to ascertain that they solved the problem in the correct manner and also to take note of areas in which the system could not understand user input.

The *briefing stage* included the test coordinator (the author) explaining the purpose of the evaluation and the activities of the evaluation. The three prototypes were also explained to the participant. Participants were assured that the prototypes were being evaluated according to their experience with each prototype and did not necessarily evaluate the participant themselves. The participants were then shown the test plan and the post-test questionnaire (Appendix C) and the purpose of these documents were explained. The consent form (Appendix C) was then given to the participants. The consent form detailed the purpose of the evaluation and also assured the participant that the data gathered would not be seen by any other person other than the person conducting the data capturing process (the author) and will not be utilised to take advantage of the said person in any way. Once the user signed the consent form the task completion phase was initiated.

The *task completion phase* of the evaluation process involved participants acting upon the case studies presented to them. The participant read each case study and upon completion started the timer and initiated interaction with one of the prototypes. Once the task was completed the time was recorded by the test coordinator. The participant then proceeded to the next case study. Once all three case studies were completed on a prototype, the same case studies were acted upon on the other two prototypes. Upon completion of all tasks on the three prototypes the test coordinator initiated the post-test questionnaire process.

In the *post-test questionnaire phase* participants answered both qualitative and quantitative questions presented to them in the questionnaire. This questionnaire evaluated the effectiveness, efficiency and satisfaction of the three prototypes.

Once the post-test questionnaire was completed by the participant a *debriefing session* was conducted. This was done to answer any question asked by the participants with regards to the post test-questionnaire. The data analysis phase consisted of analysing participants biographical details, analysing quantitative responses using descriptive and inferential statistics as well as conducting a thematic analysis of the qualitative responses.

6.3.3 Selection of Participants

Participants were chosen from a list acquired from the Help Desk at NMMU. The list contained all employees and students that had reported printer problems in a four month period. The participants were called one by one and briefed about the nature of the research and were screened to fit a certain user profile (Appendix D). The participants must have reported one of the problems that were addressed in our case studies. Twenty participants were chosen. Twelve of these were female (60%) and eight male (40%) (Figure 6.1).

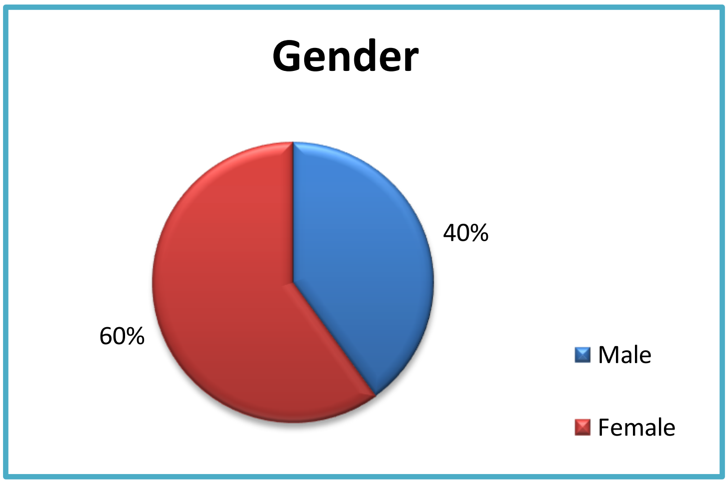

Figure 6.1 Gender Profile of Test Participants (n=20)

The majority (70%) of the participants were between 20-30 years old. The remaining 30% were either between 31-40 years or older than 50 years old (Figure 6.2).

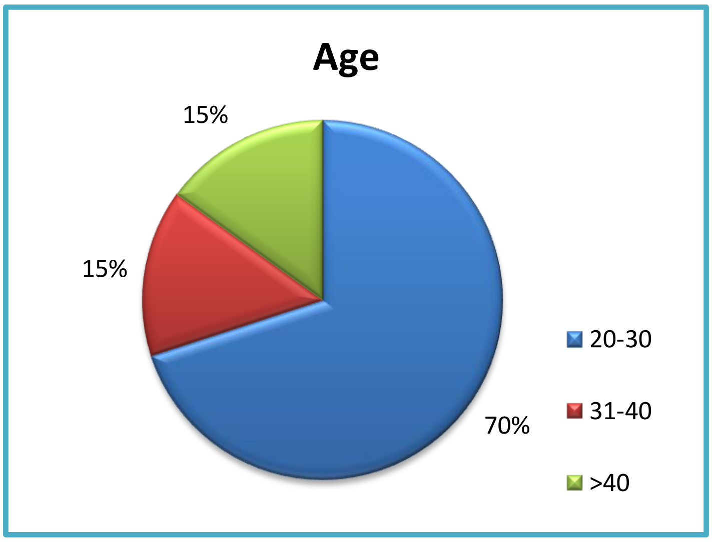

Figure 6.2 Age Profile of Test Participants (n=20)

90% (n=18) of the test participants had more than 4 years of computer experience and the remaining 10% had less than 1 year experience (n=1) and one to two years experience (n=1) (Figure 6.3).

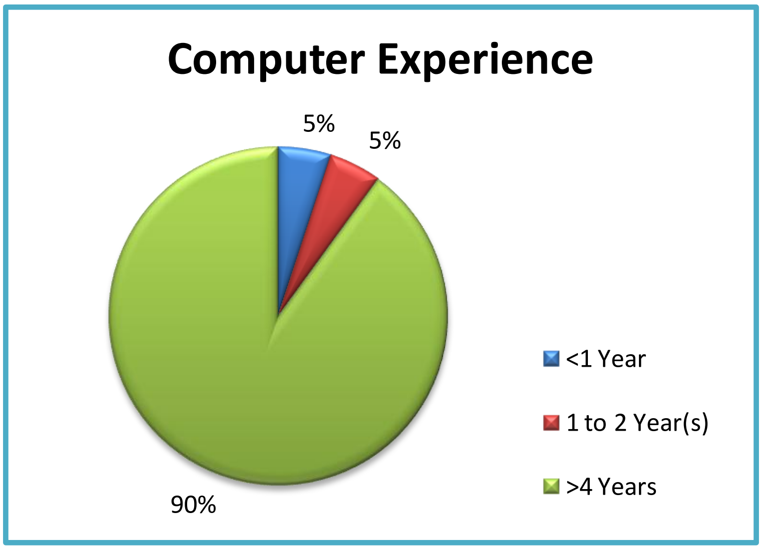

Figure 6.3 Computing Experience Profile of Participants (n=20)

75% (n=15) of participants had access to a printer and have utilised them them for greater than four years. 15% (n=3) of the participants have done the same for three to four years while the remaining participants (n=2) have only done so for less than a year (Figure 6.4).

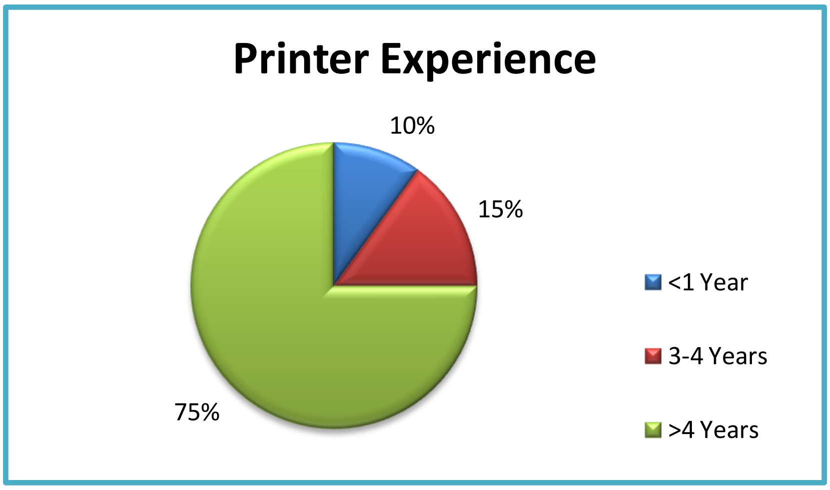

Figure 6.4 Printer Experience Profiles of Test Participants (n=20)

Participants were also asked to state their printer troubleshooting experience and 40% (n=8) stated that they had greater than four years troubleshooting experience. 5% (n = 1) stated that they had three to four years experience while 15% and 25% state that they have one to two years and less than one year of troubleshooting experience respectively. A further 15% had no prior troubleshooting experience. Figure 6.5 illustrates these statistics.

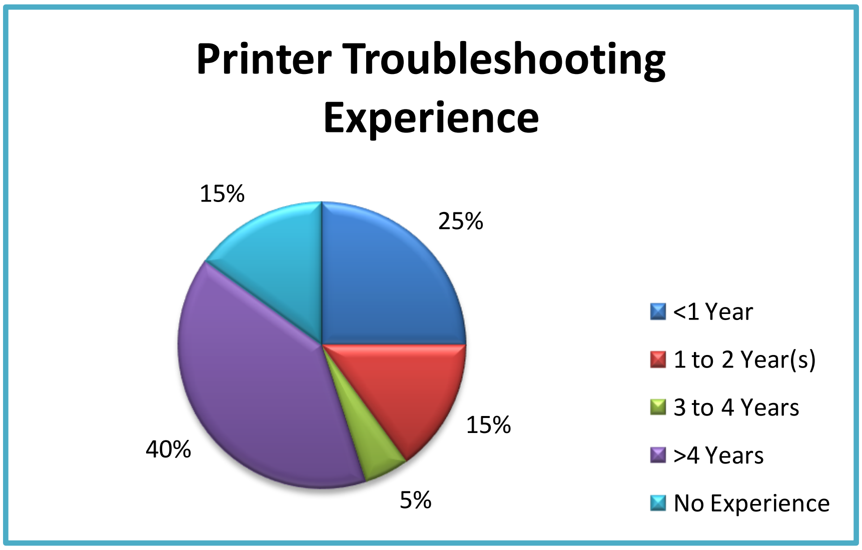

Figure 6.5 Printer Troubleshooting Experience of Test Participants (n=20)

Participants were also asked to state their instant messaging experience as the ALICE and Hybrid prototypes utilised this metaphor. 55% percent of the participants stated that they had greater than four years of experience while a further 20% stated that they had three to four years experience. The remaining 25% was split as such: 15% had 1-2 years experience and the remaining 10% had either less than one year experience or no experience at all. Figure 6.6 illustrates these statistics.

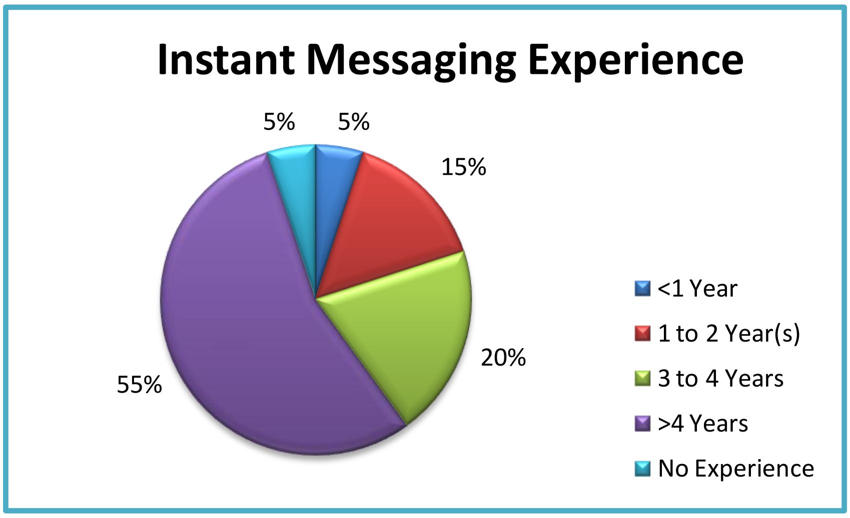

Figure 6.6 Instant Messaging Experience Profiles of Test Participants (n=20)

6.4 Results of User Testing

Further descriptive analysis of the post test questionnaire was done to determine the effectiveness, efficiency and satisfaction of the participants with the three prototypes. The post-test questionnaire yielded both qualitative and quantitative results. Detailed accounts of the quantitative results are given in Appendix E while the notable qualitative results will be discussed later on this section.

6.4.1 Quantitative Results

Ratings were given on a five point Likert scale and were rated to each usability criteria in the post-test questionnaire. Overall ratings were calculated by obtaining the mean, median, mode and standard deviation values of all participants that took part in the evaluation.

Participants were posed the questions presented in table 6.2 for all three prototypes. The fourth question was not posed for the IVR system as it is not possible to reformulate questions using this system as participants had to choose options rather than stating their queries. All questions were posed with a five point Likert scale where participants had five options to choose from. A rating above the value of three suggests that there is no cause for concern, but a rating below three means that there is cause for concern. An exact value of three suggests that the user has a neutral opinion. As was mentioned in section 6.3.1 some of the scales were reversed and during analysis these results were reversed to the correct order to maintain a standard scale. Table 6.2 highlights the questions asked for each prototype.

	Question	Hybrid	IVR	NLI
1	Did you get all the information you wanted using the system?	✓	✓	✓
2	Do you think the system understood what you asked?	✓	✓	✓
3	How easy is it to obtain the information you wanted?	✓	✓	✓
4	How easy was it to reformulate the questions when you were invited to?	✓	✗	✓
5	Do you think you would use this system again?	✓	✓	✓

Table 6.2 Questions asked in the Post-Test Questionnaire for the Three Prototypes

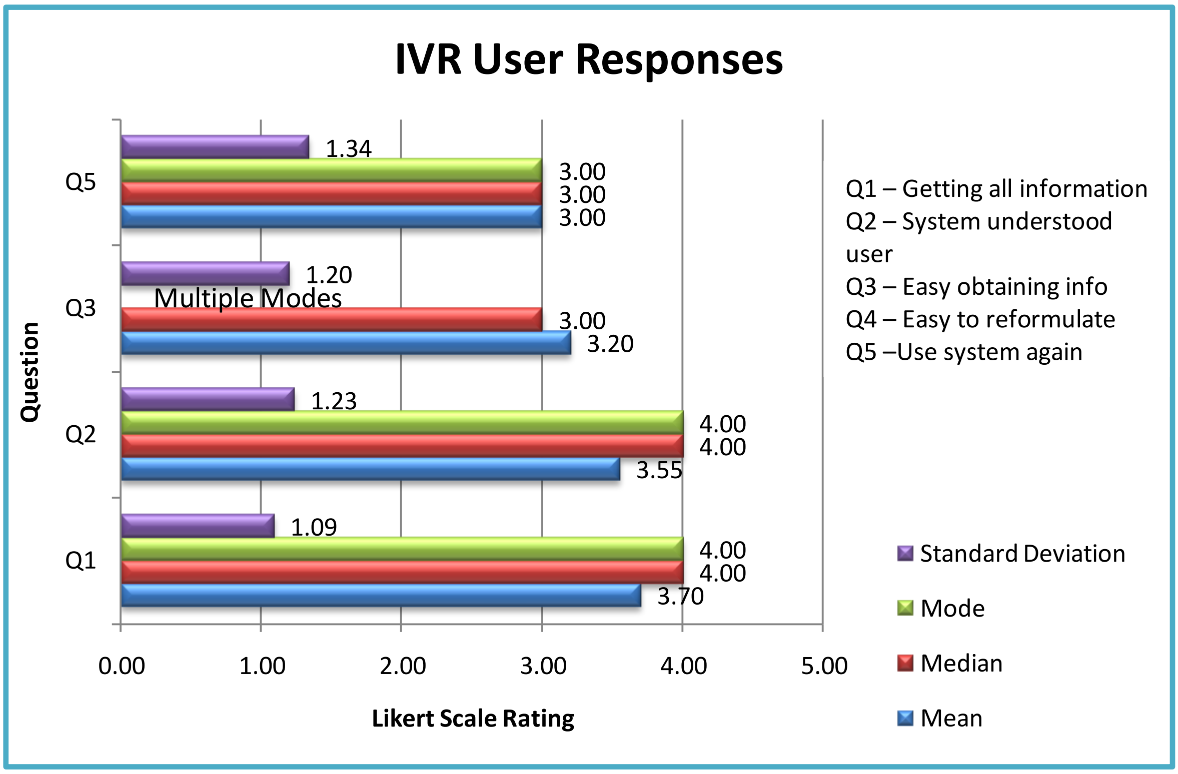

Figure 6.7 IVR Participant Quantitative Responses (n=20)

Figure 6.7 illustrates user ratings for the usability criteria specified in the post test questionnaire. The mean, median, mode and standard deviation are plotted against each other in accordance to each question posted to the user. The first question asked the participant if they got all the information utilising the system and the mean answer was 3.7 which means that participants got the answers they were looking for most of the time. The median and mode of 4 also confirms this. The second question scored a mean of just above three meaning that users felt that that the systems sometimes felt like it understood their input and the median and mode of 4 confirm this. The last two questions posted to the user have a mean of 3 and 3.2 respectively, which indicated that participants were of a neutral opinion as to whether it was easy to obtain information and whether they would use this prototype again. The medians of 3 also confirm this.

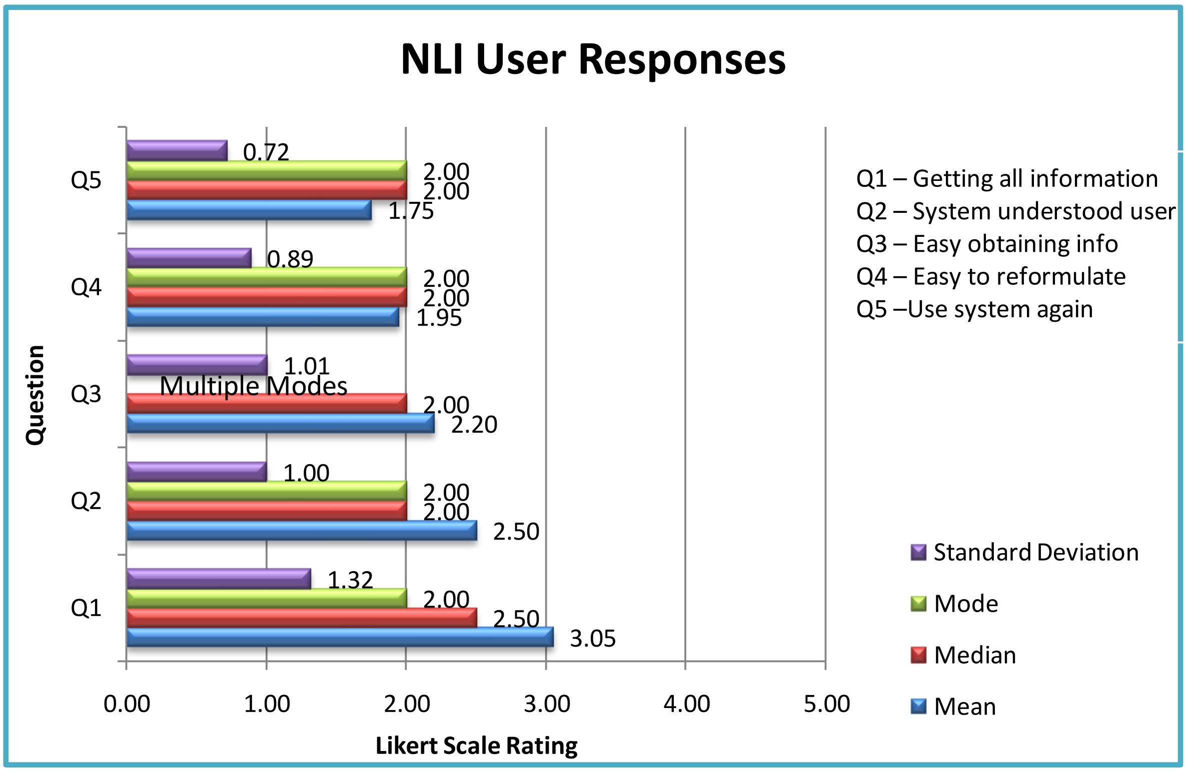

Figure 6.8 NLI Participant Quantitative Responses (n=20)

The overall participant responses gathered for the NLI (ALICE) prototype are illustrated in Figure 6.8. It can be seen that participants are of a neutral opinion as to whether the system understood them. The mean and median (2 and 2.5 respectively) scores, however; reveal that participants were not really sure if the system understood them all the time. This could be due to the fact that participants may have had to rephrase the questions to the system which is brought upon by the ambiguous nature of natural language. This can be confirmed by observing the descriptive stats for Question 4 which has a mean of 1.95 indicating that it was fairly difficult to reformulate questions. Participants also felt that it was difficult to obtain the information that they wanted and a mean score of 2.2 confirms this. Having considered the above it is understandable that users are of a neutral opinion (3.05) as to whether they were able to get all the information from the system and therefore are not too sure whether they would utilise the system again (1.95).

The overall user reaction for the Hybrid prototype is plotted in Figure 6.9. The values ranged from four and five indicating that the participants found this prototype highly effective and efficient. The standard deviations are also very small meaning that the scores were not very dispersed. The participant's indicated that it was easy to get all the information that they needed, the system understood them, it was easy to get all the information that they wanted and they were able to reformulate questions easily when it was needed. They also found the system satisfying as they said that they would use the system again.

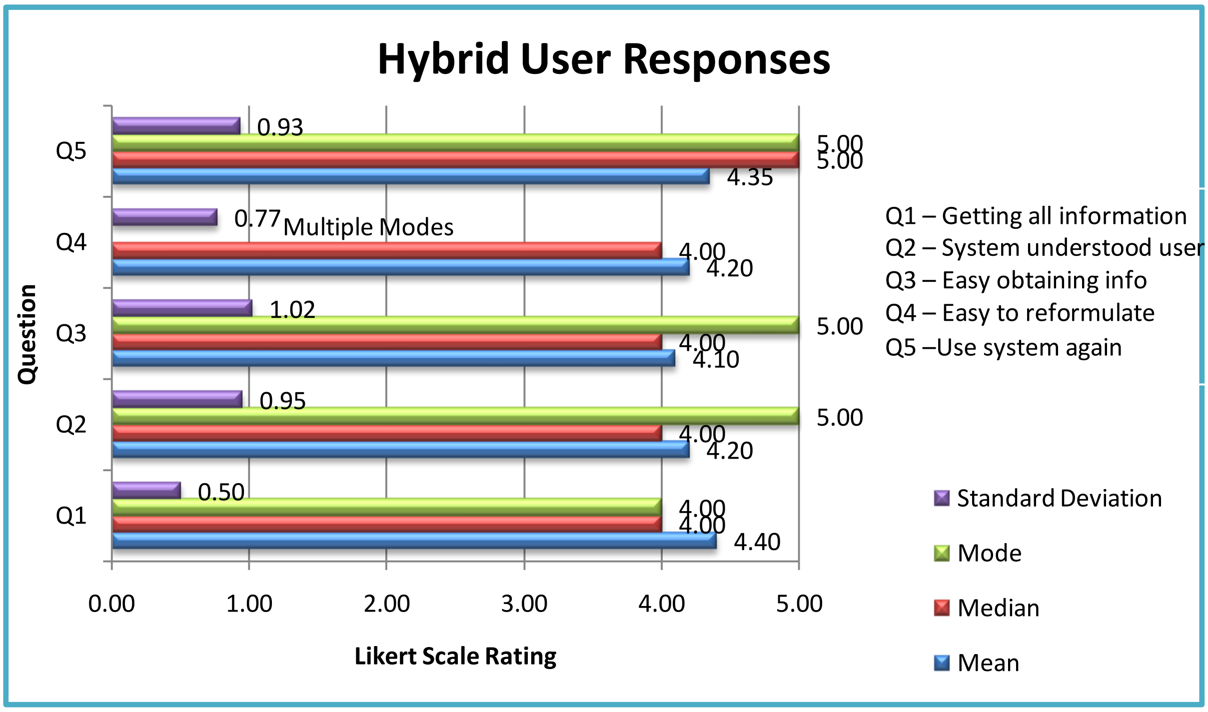

Figure 6.9 Hybrid Participant Quantitative Responses (n=20)

Figure 6.10 depicts ranking data collected from the post-test questionnaire. One of the goals of this evaluation was to determine which prototype the participants preferred. Each prototype was ranked from 1 to 3 by each participant (Appendix E, Table E.7). As can be seen from the mean rankings in Figure 6.10 the Hybrid system was ranked first by most of the participants (85%) which was the followed by IVR. ALICE was ranked third (75%) by most participants. Generally participants preferred the Hybrid system to the other prototypes.

Time on task was measured not as a measure of efficiency, but to ascertain whether this attribute had an impact on the quantitative analysis of the prototypes by the participants. Figure 6.11 indicates that the IVR system was the quickest to use while the Hybrid system was a little slower and the ALICE took the longest to utilise. This could be because participants had to only type in an option into the IVR prototype whereas with the other prototypes they had to utilise natural language to interact with them. The main reason why the Hybrid system may have been quicker than the NLI (ALICE) system is that participants may have had to rephrase a lot of their responses/queries because the system lost context during the conversation.

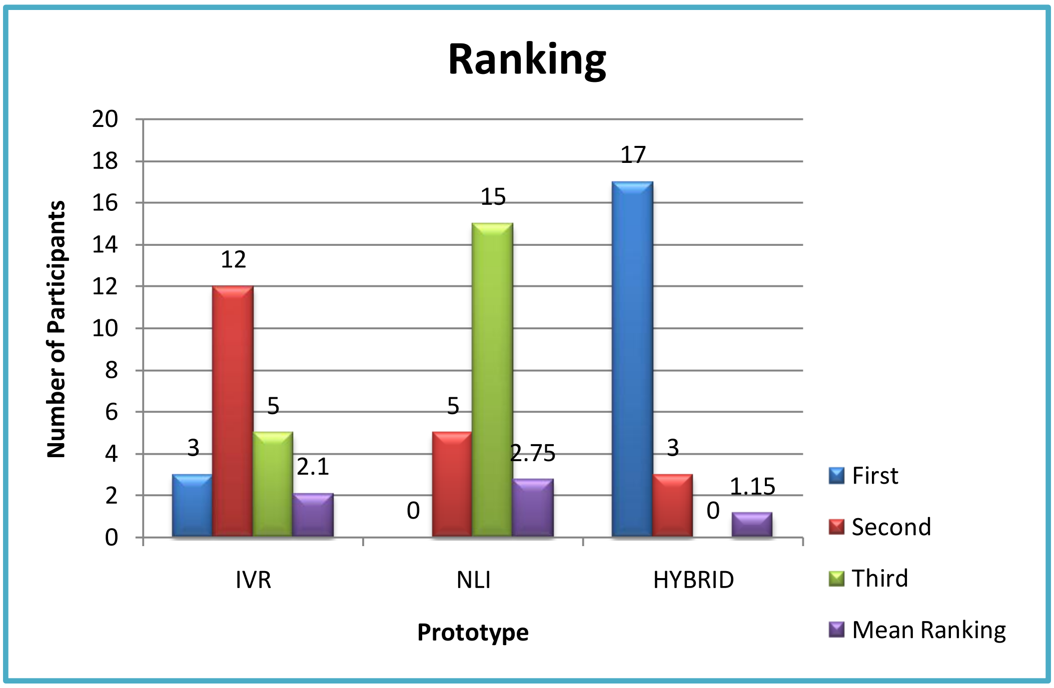

Figure 6.10 Ranking of Prototypes

The next section will highlight the inferential statistics performed on the data collected from the evaluation.

6.4.2 Inferential Statistics

Inferential Statistics are utilised to draw conclusions that cannot be done by looking at the data alone (Trochim 2006). They were firstly utilised to determine whether time on task differences on the three prototypes were statistically significant. Usability measures for the three prototypes were also calculated and compared to identify if the differences were statistically significant. Lastly inferential statistics were used to identify whether time had a direct influence on user responses for the usability metrics of the three prototypes.

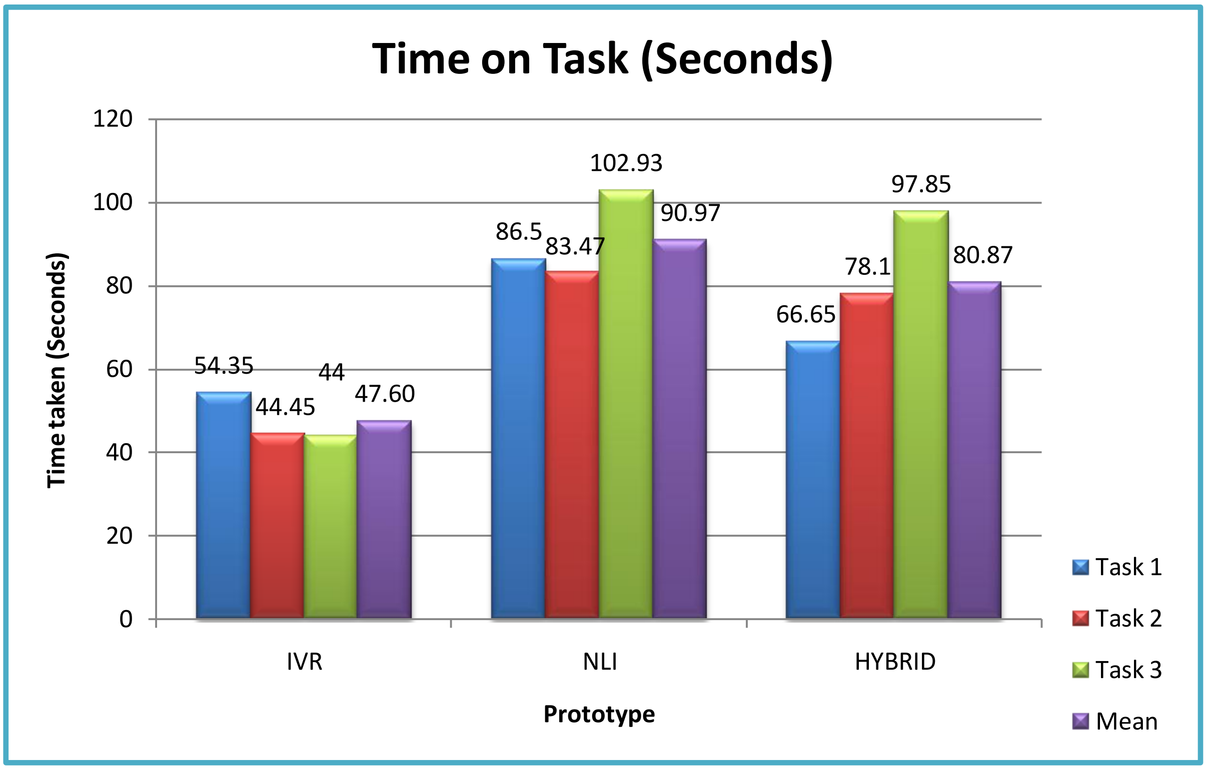

Figure 6.11 Time on Task across the Three Prototypes

6.4.2.1 Time Taken per Task

T-tests for dependant samples were conducted across each task between all prototypes in order to ascertain if there was significance in the differences in time taken to complete tasks. *T*-tests are parametric in nature and therefore assume that a normal distribution needs to present. Due to a small sample size (n=20), this could not be assumed and therefore a non-parametric test was additionally conducted to confirm the results of the t-test. The Wilcoxon matched pairs test was conducted for this purpose (Appendix E, Tables E.8 to.E.25) (Tables 6.3 – 6.5). It can be noted that there is a significant difference in time between IVR and Hybrid as indicated by tasks 2 and 3 (Tables 6.4 and 6.5). This can be explained by the fact that users only had to input a menu option in IVR and had to utilise natural language in the Hybrid system. Furthermore there were significant differences between NLI and IVR as indicated by tasks 1 and 2 (Tables 6.3 and 6.4). This can be explained by the fact that participants had to rephrase questions quite a few times for the NLI prototype.

TASK 1 across prototypes			
T-test and Wilcoxon Matched Pairs test (*T-test values in brackets)*			
	IVR **54.35 Secs**	**NLI** **86.5 Secs**	**Hybrid** **66.65 Secs**
IVR **54.35 Secs**		0.1169 (0.0585)	0.9256 (0.4188)
NLI **86.5 Secs**	0.1169 (0.0586)		***0.0196** **(*0.0119)**
Hybrid **66.65 Secs**	0.9256 (0.4188)	***0.0196** **(*0.0119)**	

Table 6.3 T-Test and Wilcoxon Matched Pairs Test p Values for Task1

(* indicates significance at the 0.05 level)

TASK 2 across prototypes			
T-test and Wilcoxon Matched Pairs test (*T-test values in brackets)*			
	IVR **44.45 Secs**	**NLI** **83.47 Secs**	**Hybrid** **78.1 Secs**
IVR **44.45 Secs**		****0.0076** (***0.046**)	****0.0046** (***0.0295**)
NLI **83.47 Secs**	****0.0076** (***0.046**)		0.1641 (0.2648)
Hybrid **78.1 Secs**	****0.0046** (***0.0295**)	0.1641 (0.2648)	

Table 6.4 T-Test and Wilcoxon Matched Pairs Test p Values for Task2

(** indicates significance at the 0.01 level, * indicates significance at the 0.05 level)

TASK 3 across prototypes			
***T*-test and Wilcoxon Matched Pairs test (*T-test values in brackets*)**			
	IVR **44 Secs**	**NLI** **102.93 Secs**	**Hybrid** **97.85 Secs**
IVR **44 Secs**		****0.001** **(**0.0009)**	****0.0001** **(**0.0002)**
NLI **102.93 Secs**	****0.001** **(**0.0009)**		0.1520 (0.9832)
Hybrid **97.85 Secs**	****0.0001** **(**0.0002)**	0.1520 (0.9832)	

Table 6.5 T-Test and Wilcoxon Matched Pairs Test p Values for Task3

(****** indicates significance at the 0.01 level)

6.4.2.2 Usability Metrics

Participant responses were categorised into the usability metrics that were measured in the evaluation: effectiveness, efficiency and satisfaction. To calculate these scores the first two questions were collapsed into one score and a mean was calculated in order to measure effectiveness (Appendix E, Table E.26). The same was done for the next two questions to measure efficiency (in the case of IVR only question three was considered) and the last question was utilised for satisfaction (Appendix E, Table E.26). It can be deduced from Figure 6.12 that the Hybrid system obtained a higher effectiveness, efficiency and satisfaction score than the other two prototypes (IVR and NLI). The repeated measures ANOVA test was conducted (Appendix E, Tables E30, E31 and E33) in order to prove statistical significances in the difference between the different usability metrics across each prototype.

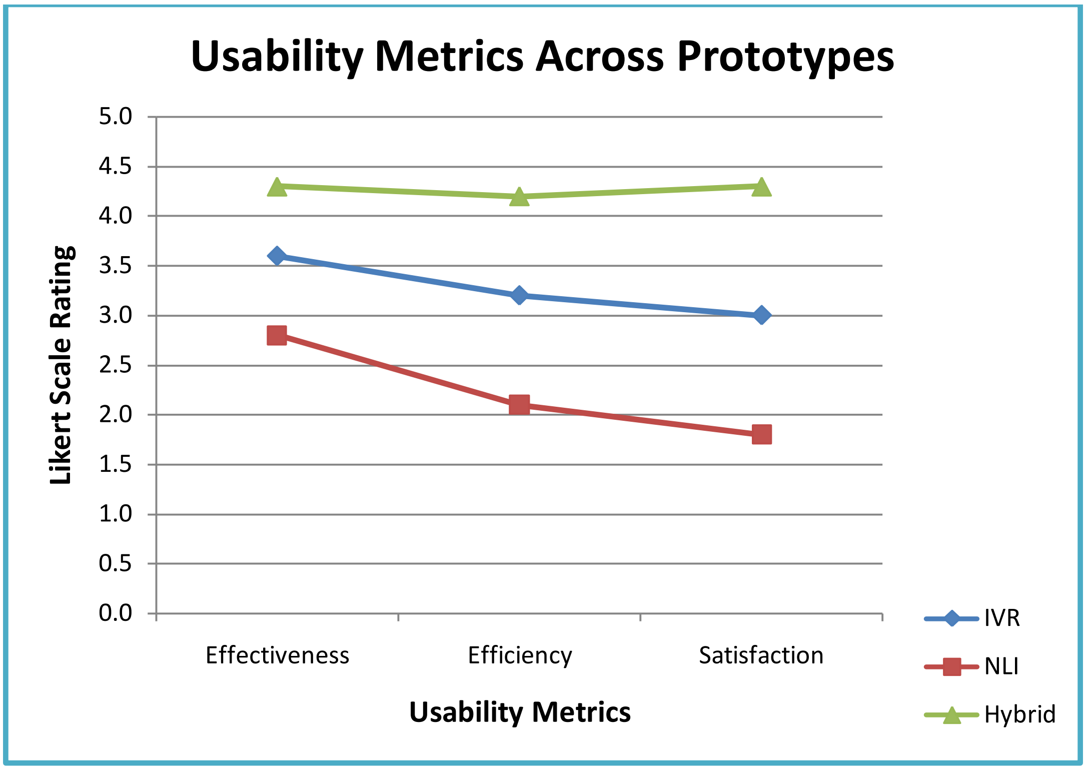

Figure 6.12 Usability of the Three Prototypes

ANOVA test for prototypes	
Prototype	p Value
NLI	****0.000019**
IVR	0.057308
Hybrid	0.612263

Table 6.6 Repeated Measure ANOVA Test p Values for Prototypes

(****** indicates significance at the 0.01 level)

Table 6.6 highlights that NLI has significant differences between the three usability measures. This needed to be investigated further to ascertain which of the three metrics had a significant difference. In order to achieve this, the Tukey honesty significant differences (HSD) test was conducted (Appendix E, Table E.32). This test is utilised for testing the significance of unplanned pairwise comparisons (Section 6.2.4).

NLI Prototype			
Tuckey HSD test **Error: Within MSE = .37346, df = 38**			
	Effectiveness +2.4450	Efficiency +2.0750	Satisfaction +1.75
Effectiveness		****0.002489**	****0.000135**
Efficiency	****0.002489**		0.225349
Satisfaction	****0.000135**	0.225349	

Table 6.7 Tukey HSD Test for ALICE Prototype

(****** indicates significance at the 0.01 level, + indicates average Likert scale score)

Table 6.7 shows that effectiveness is significantly different from efficiency and satisfaction. This could be because the participants were looking for an automation tool that could solve their problems rather than solving it as quick as possible or they were not really looking for a satisfying experience. This is also proved by the fact that efficiency and satisfaction are not significantly different.

The three usability metrics were also compared across all the prototypes (Appendix E, Tables E.34 – E.36). In other words effectiveness, efficiency and satisfaction were compared against each other in order to determine whether each usability metric was different across the prototypes. This was done to prove that all three metrics were statistically different to each other and that one prototype definitely stood out in terms of this.

Usability Metric	p Value
Effectiveness	****< 0.0**
Efficiency	****< 0.0**
Satisfaction	****< 0.0**

Table 6.8 Repeated Measure ANOVA Test p Values for Prototypes

(****** indicates significance at the 0.01 level)

The repeated measure ANOVA shows significant differences between the three usability metrics between the prototypes (Appendix E, Tables E.37 – E.39) (Table 6.8). The Tukey HSD test was conducted for each of the three metrics.

Effectiveness Metric			
Tuckey HSD test **Error: Within MSE = .37346, df = 38**			
	IVR +3.6	ALICE +2.8	Hybrid +4.3
IVR		****0.0015**	***0.0119**
NLI	****0.0015**		****0.0001**
Hybrid	***0.0119**	****0.0001**	

Table 6.9 Tukey HSD Test Showing Significant Difference between the Prototypes with the effectiveness metric

(** indicates significance at the 0.01 level,* indicates significance at the 0.05 level

+ indicates average Likert scale score)

Table 6.9 illustrates that effectiveness is significantly different across all the prototypes. Therefore from the average score across each prototype it can be deduced that the Hybrid prototype has the highest effectiveness followed by IVR then NLI.

Table 6.10 illustrates that efficiency is significantly different across all the prototypes. Therefore from the average score across each prototype it can be deduced that the Hybrid prototype has the highest efficiency followed by IVR then NLI.

Table 6.11 illustrates that satisfaction is significantly different across all the prototypes. Therefore from the average score across each prototype it can be deduced that the Hybrid prototype has the highest satisfaction followed by IVR then NLI.

Efficiency Metric			
Tuckey HSD test **Error: Within MSE = .37346, df = 38**			
	IVR +3.2	ALICE +2.1	Hybrid +4.2
IVR		****0.0016**	****0.0079**
NLI	****0.0016**		****0.0001**
Hybrid	****0.0079**	****0.0001**	

Table 6.10 Tukey HSD Test showing Significant Difference between the Prototypes with the Efficiency Metric

(****** indicates significance at the 0.01 level, * indicates significance at the 0.05 level

+ indicates average Likert scale score)

Satisfaction Metric			
Tuckey HSD test **Error: Within MSE = .37346, df = 38**			
	IVR +3.0	ALICE +1.8	Hybrid +4.3
IVR		***0.0026**	***0.0012**
NLI	***0.0026**		***0.0001**
Hybrid	***0.0012**	***0.0001**	

Table 6.11 Tukey HSD Test showing Significant Difference between the Prototypes with the Satisfaction Metric

(****** indicates significance at the 0.01 level, * indicates significance at the 0.05 level

+ indicates average Likert scale score)

6.4.2.4 Time Taken versus Usability Metrics

Once the usability metrics were calculated it was important to note how time affected the user scores on the three metrics. As mentioned before, time was not recorded as a measure of efficiency but to ascertain whether it had an impact on participants scoring in the post-test questionnaire. The Spearman rank order correlation was utilised to measure the correlation between time and the usability metrics (effectiveness, efficiency and satisfaction).

IVR			
	Task 1	Task 2	Task 3
Effectiveness	-0.21	-0.04	-0.16
Efficiency	-0.04	0.38	0.09
Satisfaction	-0.17	0.19	0.01

Table 6.12 Spearman Rank Order Correlations for Usability Metrics on IVR

NLI			
	Task 1	Task 2	Task 3
Effectiveness	-0.40	-0.30	-0.31
Efficiency	-0.22	-0.15	-0.07
Satisfaction	-0.13	-0.17	-0.14

Table 6.13 Spearman Rank Order Correlations for Usability Metrics on IVR

Table 6.12 and 6.13 relates to the IVR and NLI prototypes, and shows us that time did not have any correlation to how the participants had answered any of the quantitative questions in the questionnaire. This indicates that participants scored the IVR and NLI systems on its functionality and usability.

Hybrid			
	Task 1	Task 2	Task 3
Effectiveness	-0.42	-0.07	***0.60**
Efficiency	-0.27	-0.16	0.08
Satisfaction	***-0.46**	-0.07	0.32

Table 6.14 Spearman Rank Order Correlations for Usability Metrics on Hybrid

(* indicate significant correlations)

Table 6.14 highlights that there was a correlation between time taken and satisfaction for the Hybrid prototype. The negative correlation (-0.46) shows that the longer the participant took to complete the task, the lower their score would be for user satisfaction (negative impact). Participants therefore wanted to complete the first task as quickly as possible and would therefore prefer this prototype to the others. Figure 6.11 shows that participants took less time in the first task in comparison to the next two tasks.

When scoring the Hybrid prototype for effectiveness based on task 3 the participants gave a higher score when they took longer to complete the task (Table 6.14 has a positive correlation of 0.60). This task took longer on both the natural language interface prototypes (NLI and Hybrid see Figure 6.11). This task involved a number of steps and therefore instructions had to be clear and users had to follow them correctly in order to complete the task correctly.

6.4.3 Qualitative Analysis

An analysis was conducted on the qualitative data that was obtained from the post-test questionnaire. The qualitative data was in the form of general comments obtained from the participants. The analysis yielded both positive and negative comments.

6.4.3.1 Negative Comments

The negative comments received from the participants regarding to the IVR prototype was related to the menu structure that the system utilised. Participants felt that this was not a natural form of interaction. Some of the negative comments with regards to this were:

- *"Very time consuming – going through all the menu options"*
- *"When I choose the wrong menu option it is a pain going back to previous menus to solve the problem"*
- *"It felt longer using this system because there are too many options"*

Participants mostly pointed out that the NLI prototype was not able to sort out their problems and took far too long. Participants went further to say that the prototype was frustrating as they had to rephrase many of their requests. Some of the comments that confirm this were:

- *"Not user friendly at all—I had rewrite my questions in different ways for the system to understand what I was saying."*
- *"This system takes too long to get used to and also it does not seem to understand what I am saying to it.*
- *"Too much rephrasing"*

Participants of the Hybrid prototype mainly complained about the time it took for them to complete their tasks in comparison to IVR:

- *"Took longer than IVR. I prefer the IVR"*
- *"Good system, but I use IVR type things literally every week and have gotten used to them"*

6.4.3.2 Positive comments

Many of the participants were positive regarding the IVR prototype being quick to utilise as well as the fact that their choices were simple. Some of the comments made by the participants reflect this:

- *"Single character to type in so it was very quick"*
- *"Quick, easy and simple"*
- *"Relatively quick method"*
- *"I liked the fact that it did not require much typing and it was specific to your problem"*

The NLI prototype did not yield any positive comments from the participants. This was possibly caused by the fact that users were frustrated at the rephrasing caused by ambiguity in the prototype.

The key features that participants enjoyed in the Hybrid prototype was that it understood them better than the NLI system and that it gave informative responses to participant queries. Participants also noted that they found this prototype very user friendly. Some of the remarks made by the participants were:

- *"Understood me much better than NLI"*
- *"Good tool to solve printer problems, I would definitely use this."*
- *"Very user friendly."*
- *"Simple and straight to the point"*
- *"Simple, quick and user friendly"*
- *"Simple to use"*

Based on the analysis of the negative and positive feedback, it can be deduced that participants enjoyed utilising the IVR and Hybrid prototypes for the purposes of automation at contact centres. The comments all highlighted the effectiveness, efficiency and satisfaction of these two prototypes. The qualitative analysis, however, indicated that participants preferred the Hybrid prototype to IVR.

The previous three sections have discussed the usability of the prototypes through the qualitative and quantitative analysis of data obtained from the post-test questionnaire. The analysis has showed that participants preferred the Hybrid prototype to the other automation tools.

6.5 Summary of Results

Problem	IVR	NLI	Hybrid
Input Restriction	✗	✓	✓
Ambiguity	✓	✗	✗
Short Term Memory	✓	✗	✓
✓ ***- Problem Addressed***		✗ ***- Problem not addressed***	

Table 6.15 Summary of advantages and disadvantages of three prototypes

Table 6.15 summarises the advantages and disadvantages of NLI, IVR and Hybrid prototypes. A Hybrid prototype that utilised NLI and IVR concepts was implemented (Chapter 5) and evaluated (Chapter 6) to minimise the disadvantages and maximise advantages mentioned in Table 5.1. The evaluation yielded that the Hybrid prototype was more effective, efficient and satisfying than the other two concepts (NLI and RBS). The Hybrid prototype in essence was a natural language interface and therefore was evaluated as such. The major disadvantages of NLIs (ALICE as a case study) are that it has a short term memory and that user input can be ambiguous. The Hybrid prototype was able to address the problem of short term memory through the use of a rule base. This is highlighted in the qualitative and quantitative results of the evaluation. Ambiguity still remains a problem as it would be impossible to cater for every input that the user would present to the system.

Effectiveness measures whether a prototype can support user goals (Rogers *et al.* 2007). The Hybrid prototype scored higher than the NLI and IVR prototypes and its users were able to complete their tasks (Section 6.3.2.2). It can therefore be deduced that short term memory was not a problem in this prototype.

The main disadvantages of the IVR systems are that user inputs are restricted and menu options have to be remembered. This lead to users feeling frustrated at these systems. The Hybrid prototype utilised natural language and therefore users were able to state their queries in natural language and therefore users did not have to remember menu options. The Hybrid prototype also scored higher than the NLI and IVR prototypes in the satisfaction metric. It can therefore be deduced that users were ***not*** frustrated when they utilised this prototype.

Analysis of the results yielded that the Hybrid prototype was not as quick as the IVR in terms of task completion. Despite this the Hybrid prototype did not require too much effort to utilise as it scored higher on efficiency than the NLI and IVR prototypes.

Chapter 7 - Conclusion

I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right -- Albert Einstein

7.1. Introduction

A company's contact centre is one of the most important communication channels between that company and its customers. Customers expect a satisfying experience when using a contact centre. This means that they expect that the problems they are experiencing will be resolved efficiently and effectively by a phone call to a contact centre. If not, the company will not retain its present customers or attract new ones. Customer satisfaction is one of the aims of contact centres and in the past, the perception was that this could only be provided through friendly and knowledgeable staff.

High personnel costs have led contact centres to consider ways in which they can reduce their expenses. As a result, they employed automation techniques such as IVR which resembles a forward chaining rule-based expert system and functions in a similar manner to these systems. IVR gathers information through the use of single-worded replies or short phrases. The system utilises a tree structure (Figure 2.5), therefore every choice made by a user determines the path to be traversed in a tree. This allows the system to maintain the context of the problem and allows the user to understand the various steps taken to solve the problem. This technology has, however, come under scrutiny as customers find these systems cumbersome to use, resulting in a low resolution of queries.

An NLI is an interface that takes natural language in the form of speech or text as input to interact with the user. It has the ability to flatten or even eliminate the menu structure found in IVR. This would be because users would be able address the system in natural everyday language and would not be restricted to input in the form of choices as is the case with IVR. NLIs do have their disadvantages of ambiguity and loss of context.

Two prototypes which resembled an IVR and NLI were implemented to confirm the advantages and disadvantages of these techniques. Hybrid models were proposed which combined the concepts of an NLI and an RBS. The two models were the NLI that uses RBS concepts and a RBS that uses an NLI. The first model was chosen and the main reason for the choice was that it could be implemented by anyone who understood the domain well with very little training. This model also directly answers the primary research question stated in Chapter 1.

The implementation process was described in Chapter 5. The main implementation issues that were encountered were the permutations and combination of rules to be tagged to the patterns in AIML and the ambiguity of natural language. A pilot study was conducted to determine whether the prototype ran smoothly. Once the bugs found in the prototype were addressed, the final evaluation was conducted with the aim of deducing the most usable prototype (effective, efficient and satisfying). The Hybrid model was ranked as number one by most of the participants and this was confirmed through descriptive and inferential statistics.

The objective of this chapter is to reflect on the research contributions (Section 7.2), to identify limitations and implications of the research (Section 7.3) as well as to make recommendations for future research (Section 7.4)

7.2. Research Contributions

The research objectives were identified in Section 1.3.1. This section states each objective together with an explanation as to how each objective was met.

The objectives of this research were identified as follows:

- *Identification of challenges to contact centres and limitations of their automation techniques with special attention paid to IVR*. This was achieved through an extensive literature study. IVR was seen as advantageous as it decreased contact centre costs. This was, however, done at the cost of decreasing customer service.
- *Identification of benefits and shortcomings of rule based expert systems. The objective was met through conducting a literature study*. IVR was compared to a forward chaining rule-based expert system. The main benefit of an RBS was identified as the reduction in employee training and personnel costs while the main shortcoming was that the system would not be able to recognise when no answer is available.
- *Identification of the different types of NLIs, their benefits and shortcomings and how they can be utilised to address the limitations of IVR*. An extensive literature study was utilised to meet this objective. Two types of NLIs were discussed namely text based and conversational speech interfaces. The main limitation of an NLI was stated as being ambiguity while the benefit was that users would be able to use natural language to address the system.
- *Implementation and evaluation of two prototypes modelled to resemble an IVR and NLI to confirm advantages and disadvantages according to theory (Chapter 5)*. This objective was achieved through the implementation of an NLI (using ALICE concepts) and an IVR. These two prototypes were then evaluated using case studies and post-test questionnaires as instruments for the evaluation. The evaluation confirmed the advantages and disadvantages in accordance to theory.
- *Propose models for a Hybrid prototype that would combine NLI and IVR concepts and implement a model*. The Hybrid prototype was proposed through the aid of two models that were proposed by the author. The NLI with RBS concepts model was chosen due to its simplicity. This model also directly answered the primary research question discussed in Chapter 1.
- *Evaluate the three prototypes (NLI, IVR and Hybrid) to determine which one is more effective, efficient and satisfying*. The model was implemented and evaluated against the NLI and IVR prototype and through descriptive and inferential statistics it was proved that the Hybrid prototype was the most effective, efficient and satisfying prototype.

The primary research question was stated in Chapter 1 as:

Can a natural language interface address the limitations and enhance the benefits brought about by IVR as an automation technique at contact centres?

This question has been answered through the use of an evaluation that compared an IVR, NLI and a Hybrid prototype. The evaluation yielded that the Hybrid prototype (which is an NLI with RBS concepts) was more effective, efficient and satisfying than the IVR.

The Hybrid prototype addressed the problem of input restriction posed by IVR as users could state their queries in natural language. This eliminated the menu structure found in IVR systems and users were able to have more natural interactions.

The benefit of a long term memory found in IVR was implemented in the Hybrid prototype through the use of a rule base. The rule base provided the Hybrid prototype with basic structure for conversations and was able to determine the next conversation to hold with the user given the current data that the prototype had gathered. The main disadvantage for the Hybrid prototype, which is an NLI, is ambiguity. This is due to the fact that it would be difficult to determine every input a user can provide. This problem was partially addressed by holding clarification dialogues in order to gather information needed by the prototype. Table 7.1 summarises how the Hybrid prototype addressed the limitations and enhanced the benefits of IVR.

Problem	IVR	NLI	Hybrid
Input Restriction	✘	✓	✓
Ambiguity	✓	✘	✘
Short Term Memory	✓	✘	✓
✓ ***- Problem Addressed***		✘ ***- Problem not addressed***	

Table 7.1 Summary of Limitations and Benefits Addressed by the Three Prototypes

Through the achievement of the research objectives and answering of the primary research question, this study has contributed to both theoretical (Section 7.2.1) and practical applications of automation at contact centres (Section 7.2.2).

7.2.1 Theoretical Contributions

The first theoretical contribution of this research is the confirmation of the advantages and disadvantages of NLI and IVR through the implementation and evaluation of two prototypes. The NLI prototype was criticized for the number of times that users had to rephrase their queries to the system due to ambiguity in natural language. The participants also stated that the conversations at times did not make sense as the NLI tended to lose focus of the problem at hand. This was caused by the lack of long term memory in NLI (ALICE). Participants, however, enjoyed their experiences somewhat as the system was able to accept natural everyday language in comparison to the menu options that they are restricted to when using IVR.

The IVR prototype was criticized for the time it took participants to complete their tasks and they found the menu structure in this system very frustrating. The participants also stated that there was too much information directed at them, which added to their frustration. Table 7.2 summarises the advantages and disadvantages of NLI and IVR.

Natural Language Interface (ALICE)	Rule Based Expert System (IVR)
✓ Users can use natural language to state their Queries.	✗ Users input are restricted to menu options.
✗ Input can be ambiguous as user input is unlimited.	✓ Ambiguity is not a problem as user input is restricted.
✗ System has a short term memory and poses a problem for users.	✓ System has a long term memory.
✓ Users input not limited by menu choices	✗ Users have to remember menu options.
✓ NLIs allow for a more natural interaction with the system.	✗ System does not feel natural as input is restricted.
✓ **Advantage**	✗ **Disadvantage**

Table 7.2 Summary of Advantages and Disadvantages of NLI and IVR

It was concluded that a Hybrid model combining NLI and RBS concepts would enhance automation at contact centres. Consequently two models which combined the concepts of NLI and IVR were proposed.

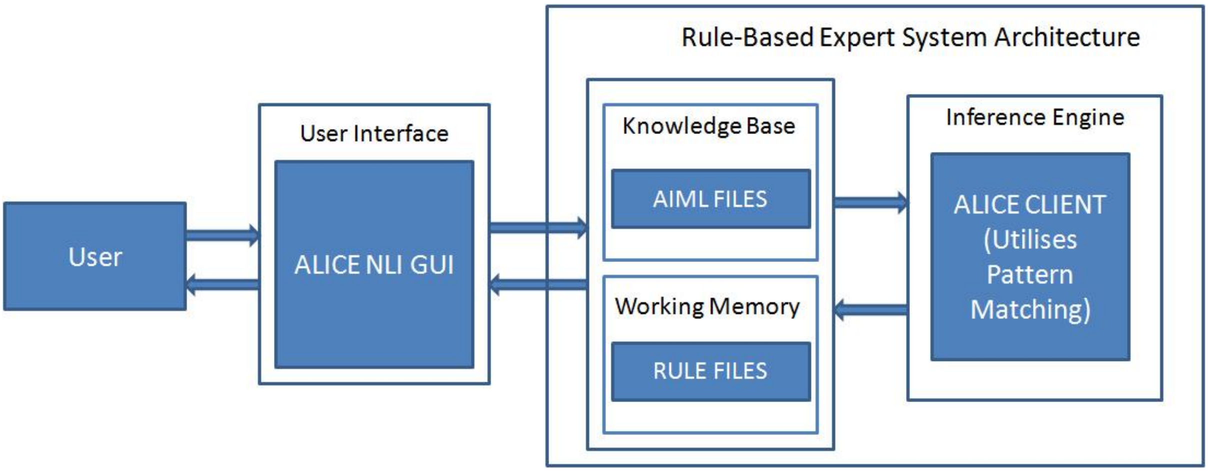

Figure 7.1 NLI using RBS concepts Model

The NLI using RBS concepts model (Figure 7.1) was chosen for implementation as it would be a more cost effective solution for contact centres and because it directly answers the primary research question (Section 7.2). Furthermore, anyone who understands the domain well would be able to implement the model with very little training.

The model consists of two main components namely the *user interface* and the *rule based expert system architecture*. The *RBS architecture* consists of three components namely the knowledge base, working memory and inference engine. The ***knowledge base*** consists of AIML files which are used by the inference engine (ALICE client). The ***working memory*** consists of rule files which store all current rules utilised by the system. The ***inference engine*** utilises ALICE pattern matching in order to produce output to user input (provided in natural language). The *user interface* is utilised to provide the user with feedback in the form of system output.

7.2.2 Practical Contributions

The practical contribution of this research consists of the implementation of a Hybrid model. An evaluation was conducted to confirm the practicality of this model. Rules for the Hybrid prototype were implemented utilising the menu structure found in the IVR prototype. Figure 7.2 highlights how the menu structure found in IVR was used to create rules. Rules were established at every user decision point, which was every time the user had to make a choice for his/her input.

The AIML was then tagged with the rules by utilising a custom AIML predicate called <Rule> (Figure 7.3). This <Rule> tag guides the system by directing it to the exact branch that must be traversed in the tree.

The <rule> predicate uses the information inside its tag to post tag the rule to the next pattern. These rules will then give the conversations structure and will be able to solve the problem in a structured manner.

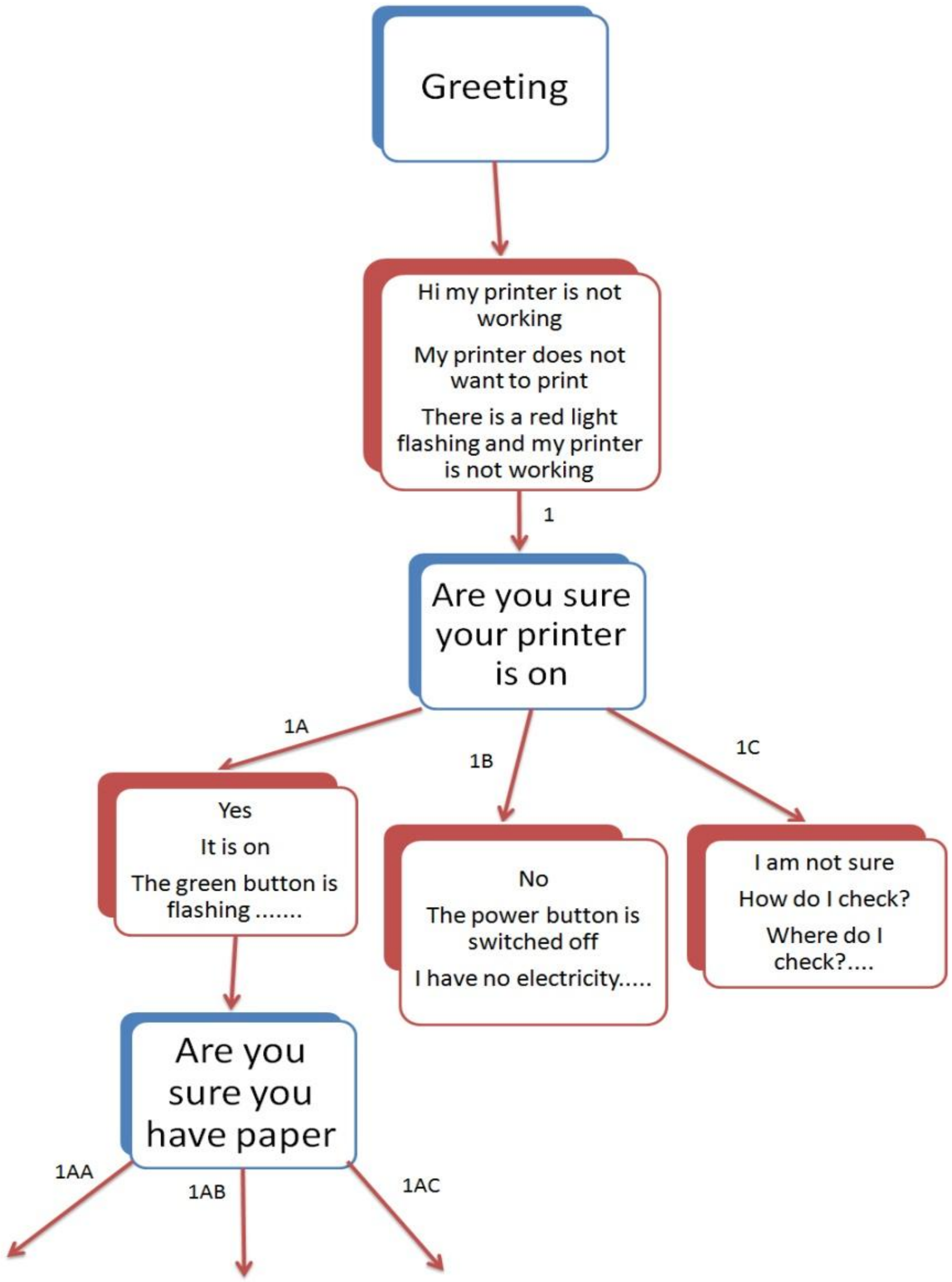

Figure 7.2 Menu Structure Utilised to Create Rules

Once the Hybrid prototype was implemented an evaluation was conducted to confirm the practicality of the model and to determine if this prototype was more effective, efficient and effective than the IVR and NLI prototypes. The evaluation through the use of inferential and descriptive statistics yielded that the Hybrid prototype was the most effective, efficient and satisfying.

```
<category>
        <pattern>
                MY PRINTER STOPPED PRINTING MY DOCUMENTS
        </pattern>
        <template>
                Are you sure your printer is on?
                <rule>1</rule>
        </template>
</category>
<category>
        <pattern>
                HOW DO I CHECK IF IT IS ON (1)
        </pattern>
        <template>
                Please locate the power button on your printer. Once located see if it is in the "on" position. Is it
in the "On" position?
                <rule>1A</rule>
        </template>
```

Figure 7.3 Custom Tags as used in the Hybrid Prototype

7.3 Implications of Research

The successful implementation and evaluation of the Hybrid prototype implies that this approach can be utilised as a more effective, efficient and satisfying automation tool at contact centres. The evaluation through descriptive and inferential statistics has proved that the Hybrid prototype is more usable than IVR.

A practical implication of this research is that a hybrid model was implemented and was highlighted as a cost saving option due to its uncomplicated implementation techniques.

7.4 Future Research

Various possibilities for future work are envisaged.

7.4.1 Implementation of RBS using NLI Concepts Model

One of the achievements of this research was the proposal of two hybrid models which utilised NLI and RBS concepts. This research concentrated on natural language interfaces therefore only the one model (NLI using RBS concepts) was implemented. Although the evaluation showed that this prototype was effective, efficient and satisfying, there is still an opportunity to implement the RBS using NLI concepts model. The implementation and successful evaluation of this model would validate it as another hybrid approach that could be utilised as an automation technique at contact centres.

7.4.2 Speech Recognition and Synthesis

The final evaluation conducted in this research yielded that the IVR prototype was quicker to utilise than the NLI and Hybrid prototypes. This was attributed to the fact that there was minimal input required by the user for the IVR to function in comparison to the user typing out sentences in natural language for the NLI and Hybrid prototypes. The use of speech recognition and synthesis could address this problem with NLI systems. Such a system would have to be evaluated in order to conclude its usefulness.

7.4.3 Embodied Conversational Agents

An embodied conversational agent is a conversational agent that has a body structure. This is usually in the form of a face that can yield facial expressions. Research has shown that these types of agents provide a more human experience to users as they are able to gather the mood of the conversations through the facial expressions of the agent. An evaluation as to whether this additional mechanism could improve user satisfaction could be conducted.

7.4.4 Other Techniques to Maintain Context of Conversation

Various other techniques that could be utilised to maintain the context of a conversation in an NLI can be researched. The literature study highlighted Bayesian networks as one possibility. Implementing another successful case study would further enhance NLI and would further push these interfaces to be considered as a standard at contact centres.

7.4.5 Implementation of the Models on a Mobile Phone

The models were evaluated on personal computers. The fast adoption of mobile phones could serve as another platform that could be utilised in the implementation of the hybrid models. Mobile phones present their own unique set of difficulties with the main one being their small screens. Implementing the models on another platform would further validate these models. Validating these models on mobile phones would be done through an evaluation which would compare the model on a personal computer to one on a mobile phone.

7.4.6 Automating the Generation of AIML with Rules

The generation of AIML to maintain context was tedious in nature as it had to be done manually (Section 5.5.3). If this process could be automated in such a way that conversations can be analysed and tagged, it would save time in the development process.

7.5 Summary

The wide use of IVR as an automation technique at contact centres has forced them to come under scrutiny for their poor quality of interactions with users that have led to a low resolution of queries. The main reason for this scrutiny is due to the poor interaction techniques that these systems employ. This research has highlighted the positive impact a natural language interface, in particular a conversational agent, could have.

An NLI has the ability to flatten and even eliminate the menu structure found in IVR, resulting in more natural interaction between the system and the user. This research has shown that, by making use of RBS concepts, an NLI, in particular a conversational agent can be given a longer term memory. This Hybrid approach makes an important contribution towards addressing current shortcomings to automation techniques employed at contact centres.

References

ABRAHAM, A. (2005): *Rule-bases Expert Systems*. In *Handbook of Measuring System Design*. SYDENHAM, P.H. and THORN, R. (eds). John Wiley & Sons Ltd.

ACAPELA GROUP (2008): *How Does Text to Speech Work* [online]. Available at http://www.acapela-group.com/how-does-text-to-speech-work.html [Accessed on 20th October 2008]

ALMSKOG & FRYDMAN COMMUNICATIONS (1996): *CALL CENTER 2000: THE FUTURE IS HERE*. *DMA TELEPHONE MARKETING*. http://www.callcentres.com.au/DMA2000.htm

ANDROUTSOPOULOS, I. (2002): *Exploring Time, Tense, and Aspect in Natural Language Database Interfaces*. John Benjamins Publishing Company.

AUSTRALIAN NATIONAL AUDIT OFFICE (1996): *Telephone Call Centres* [online]. Available at http://www.anao.gov.au/WebSite.nsf/Publications/4A256AE90015F69B4A2569120006FCD1 [Accessed on February 27 2006]

BADIRU, A.B. and CHEUNG, J. (2002): *Fuzzy Engineering Expert Systems with Neural Network Applications*. John Wiley and Sons.

BATES, R.J. and GREGORY, D.W. (2001): *Voice and Data Communications Handbook*. McGraw-Hill Professional.

BERGEVIN, R. and WYATT, A. (2008): *Contact Centres for dummies*. 2nd Edn, Wiley Publishing.

CALL CENTRE INDIA (2007): *Inbound Call Centres* [online]. Available at http://www.call-centers-india.com/inbound-call-centre.html [Accessed on 12 March 2007]

CALLTROL (2008): *What is Automatic Call Distribution?* [online]. Available at http://www.calltrol.com/automation/index_acd.html [Accessed on 12th May 2008]

CASSELL, J., STEEDMAN, M., PELACHAUD, C., STONE, M., DOUVILLE, B., PREVOST, S. and ACHORN, B. (1994): *Modeling the Interaction between Speech and Gesture*. University of Pennsylvania. Philadelphia, PA 19104-6389.

CHATBOT LIST (2007): *List of Chatbots* [online]. Available at www.chatbotlist.org [Accessed on 12th September 2008]

COHEN, P.R. (1992): The role of natural language in a multimodal interface. In *Proceedings of 5th Annual ACM Symposium on User Interface Software and Technology*, Monteray, California, United States of America. 143-149, ACM (ed) ACM Press

COHEN, P.R., DALRYMPLE, M., TYLER, S.W., MORAN, D.B., PEREIRA, F.C.N., SULLIVAN, J.W. and GARGAN, R.A. (1989): Synergistic Use of Direct Manipulation and Natural Language. In *Proceedings of Chi*. ACM (ed) ACM.May 1989.

CONTACT CENTRE UNIVERSITY (2000): *Glossary of Contact Centre Terms* [online]. Available at http://www.ctifourm.com/technology/tcs/glossary/pdf [Accessed on 3rd March 2008]

DAWSON, K. (2003): *The Call Center Handbook: The Complete Guide to Starting, Running, and Improving Your Call Center*. Focal Press.

DEWET, F., DEVETH, J., CRANEN, B. and BOVES, L. (2003): *Automatic Speech Recognition in Adverse Acoustic Conditions* Computer Science, Katholieke Universitiet Nijmegen.

DIMENSION DATA (2006): *Merchants Global Contact Centre Benchmarking Report*.

DUNN, D.S. (2001): *Statistics and Data Analysis for the Behavioral Sciences*. New York, NY 10020, McGraw Hill Higher Education.

DURKIN, J. (1994): *Expert Systems Design and Development*. Englewood Cliif, New Jersey, Prentice Hall.

DUSAN, S. and FLANAGAN, J. (2004): *Advances in Natural Multimodal Dialogue Systems*. Dordrecht, Kluwer Academic Publishers.

DUTOIT, T. (1997): *An Introduction to Text-to-speech Synthesis*. Norwell, MA, USA, Kluwer Academic Publishers.

http://books.google.co.za/books?id=xjXU3FujWbwC&dq=speech+synthesis&lr=&source=gbs_summary_s&cad=0

FEIGENBAUM, E.A. (1982): *Knowledge Engineering in the 1980's*. California, Department of Computer Science, Stanford Univeristy.

FORSYTH, R. and NAYLOR, C. (1991): *The hitch-hiker's guide to artificial intelligence*. Taylor & Francis.

FRANTZ, M.G. (2006): *The Evolution of the IT Help Desk to the Service Desk*. http://www.compucom.com/wwwimages/Corp/Images/CompuComServiceDeskWhitePaper0106.pdf

FRIEDMAN, T. (2001): *Call Centre Management: Balancing the Numbers*. *Industrial Management*:43

FUM, D., GUIDA, G. and TASSO, C. (1988): A Distributed Multi-Agent Architecture for Natural Language Processing. In *Proceedings of International Conference On Computational Linguistics*, Budapest, Hungary 2:812 - 814, Association for Computational Linguistics

FUSION (2004): *Improving contact center effectiveness and agent performance through process integration*. http://www.jacada.com/WhitePapers/pdfs/Frost-Sullivan.pdf

GANS, N., KOOLE, G. and MANDELBAUM, A. (2003): *Telephone Call Centres: Tutorial, Review and Research Prospects*.

GIARRANTANO, J. and RILEY, G. (1989): *Expert Systems: Principles and Programming*. Boston, Massachusetts, PWS-Kent.

GRZYMALA-BUSSE, J.W. (1991): *Managing Uncertainty in Expert Systems*. Vol. 1. Springer.

HAYES-ROTH, F. (1985): *Rule-Based Systems*. *Communications of ACM* 28(9).September 1985.

HENDRIX, G.G. (1982): *Natural Language Interfaces*. *American Journal of Computational Linguistics* 8(2).April-June 1982.

HILL, D.R. (1971): *Man-Machine Interface Using Speech*. In *Advances in Computers*. RUBINOFF, M. and YOVITS, M.C. (eds). Academic Press.

HILL, I. (1983): *Natural language versus computer language*. In *Designing for Human-Computer Communication*. SIME, M. and COOMBS, M. (eds). Academic Press. .

HOLMES, J. and HOLMES, W. (2001): *Speech Synthesis and Recognition*. CRC Press. http://books.google.co.za/books?id=guk_VI2UICkC&dq=speech+synthesis&lr=&source=gbs_summary_s&cad=0

ITIL RESOURCES (2008): *Service Desk Objectives in ITIL Foundation* [online]. Available at http://www.itilfoundation.org/Service-Desk-Objectives-in-ITIL-Foundation_43.html [Accessed on 19 July 2008]

JELINEK, F. (1997): *Statistical Methods for Speech Recognition*. MIT Press. http://books.google.co.za/books?id=1C9dzcJTWowC&dq=speech+recognition&source=gbs_summary_s&cad=0

JONES, C. (2005): *What is a Service Desk: Frequently Asked Questions (FAQ) #2* [online]. Available at http://connect.educause.edu/blog/ckckjones/whatisaservicedeskfrequen/75?time=1222270240 [Accessed on 26 June 2008]

KAMINSKI, D.M. (1989): A knowledge-based approach to phonetic speech recognition for the speech impaired. In *Proceedings of Third international conference on human-computer interaction on Designing and using human-computer interfaces and knowledge based systems* Boston, Massachusetts, United States 712 - 719

Elsevier Science Inc

KANDEL, A. (1991): *Fuzzy Expert Systems*. CRC Press.

KARAT, C.-M., VERGO, J. and NAHAMOO, D. (2003): *Conversational Interface Technologies*. In *The Human-computer Interaction Handbook*. 168-184. JACKO, J.A. and SEARS, A. (eds). Hillsdale, NJ, USA, Lawrence Erlbaum Associates.

http://books.google.co.za/books?id=u9zFGT8GrV4C&pg=PA176&dq=speech+synthesis&lr=&ei=bggPSdKIHYXWsgOi3pnNDg#PPA169,M1

LAI, J. and YANKELOVICH, N. (2003): *Conversational Speech Interfaces*. In *The Human-computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. 689-721. JACKO, J.A. and SEARS, A. (eds). Hillsdale, NJ, USA, Lawrence Erlbaum Associates. http://books.google.co.za/books?id=u9zFGT8GrV4C&pg=PA185&dq=Natural+Language+Interfaces&lr=&ei=D70MSe7OGYf0sQOFs8npBA&client=firefox-a#PPA699,M1

LARSON, J.A. (2005): *Regaining Consumer IVR Confidence: Companies Draw on Different Tools to Save IVR/Speech Interaction* [online]. Available at http://www.speechtechmag.com/Articles/PrintArticle.aspx?ArticleID=30532 [Accessed on 15th October 2008]

LEE, S.-I., SUNG, C. and CHO, S.-B. (2001): *An Effective Conversational Agent with User Modeling based on Bayesian Network*. Seoul, Yonsei University.

LEE, L. (2004): *I'm sorry Dave, I'm afraid I can't do that": Linguistics, Statistics, and Natural Language Processing circa 2001*. Cornell University.

LESTER, J., BRANTING, K. and MOTT, B. (2004): *Conversational Agents*. Practical Handbook of Internet Computing. www.bradfordmott.com/papers/Lester-2004.pdf

LIBERMAN, H. (2001): *Interfaces that give advice*. Human-Computer Interaction in the New Millenium. ACM Press/Addison-Wesley Publishing Co.

LONG, B. (1994): *Natural Language as an Interface Style* [online]. Available at http://www.dgp.toronto.edu/people/byron/papers/nli.html [Accessed on 20th February 2007]

LUCAS, P. and GAAG, L.V.D. (1991): *Principles of expert systems*. Kent, 1st Edn, Addison-Wiley Publishers Ltd.

MATTHEWS, J. (2002): *How Does Speech Recognition Work?* [online]. Available at http://www.generation5.org/content/2002/howsrworks.asp [Accessed on 25th September 2008]

MCBREEN, H., ANDERSON, J. and JACK, M. (2000): *Evaluating 3D Embodied Conversational Agents In Contrasting VRML Retail Applications*. Edinburgh, Centre for Communication Interface Research, University of Edinburgh.

MICROSOFT (2008): *Service Management Functions: Service Desk* [online]. Available at http://technet.microsoft.com/smfsvcdk.mspx [Accessed on 29 September 2008]

MIDDLETON, I. and MARCELLA, R. (1996): *Key Factors in Help Desk Success* The British Library.

MUNTEANU, C. and BOLDEA, M. (2000): MDWOZ: AWizard of Oz Environment for Dialog Systems Development. In *Proceedings of Conference on Language Resources and Evaluation - LREC*, Athens, Greece.

MURPHY, K. (1998): *A Brief Introduction to Graphical Models and Bayesian Networks* [online]. Available at http://www.cs.engr.uky.edu/~dekhtyar/dblab/resources.html#bnets [Accessed on 10th October 2008]

NANODESK TECHNOLOGIES (2002): *QuickSmart Help Desk Introduction Guide*.14.

OGDEN, W.C. and BERNICK, P. (1996): *Using Natural Language Interfaces*. New Mexico, New Mexico State University.1-34.

PATRIDGE, D. (1991): *A New Guide to Artificial Intelligence*. Intellect Books.

PETROVIC, S. (2000): *Rule-Based Expert Systems* [online]. Available at http://www.cs.nott.ac.uk/~sxp/ES3/index.htm [Accessed on 15th March 2008]

PURUSHOTHAMAN, S.V. (2004): *What are the benefits of using IVR technology in the call center of a contemporary organization?* [online]. Available at http://www.crm2day.com/experts/EplAklZklFJDOuHGzP.php [Accessed on 12th March 2007]

RAVI SANKAR, G., GREYLING, J. and VOGTS, D. (2008a): *Towards a Conversational Agent for Contact Centres*. *South African Telecommunications, Networks and Applications*

Conference (SATNAC). BARNARD, D., BARNARD, L., DEVILLIERS, P. and GOOD, R. Wild Coast, Eastern Cape, South Africa, Telkom. 1.

RAVI SANKAR, G., GREYLING, J., VOGTS, D. and DU PLESSIS, M.C. (2008b): *Models towards a Hybrid Conversational Agent for Contact Centres*. *South African Insititute of Computer Scientists and Information Technologists (SAICSIT)*. CILLIERS, C., BARNARD, L. and BOTHA, R. Wilderness Beach Hotel, Wilderness, George, Western Cape, South Africa, ACM. 1:200-210.

ROBERTSON, J. (2005): *Two types of in-bound call centres* [online]. Available at www.steptwo.com.au [Accessed on 20th January 2008]

ROGERS, Y., SHARP, H. and PREECE, J. (2007): *Interaction Design: Beyond Human-Computer Interaction*. The Atrium, Southern Gate, Chichester, West Sussexss, 2nd Edn, John Wiley & Sons, LTD.

SALUJA, A.S. (2006): *A Wake-up "Call" for Your IVR* [online]. Available at http://www.inductis.com/who_are_we/articleandwhitepaper/a_wake_up_call.html [Accessed on 13th June 2007]

SANDERSON, S. (2003): *Introduction to Help Desk Concepts and Skills*. McGraw-Hill. http://www.mhprofessional.com/product.php?isbn=007821677X&cat=&promocode=

SANTEN, P.H.V., SPROAT, R.W., OLIVE, J.P. and HIRSCHBERG, J. (1997): *Progress in Speech Synthesis*. Berlin, Springer.

SCHROEDER, M.R. (2004): *Computer Speech: Recognition, Compression, Synthesis : with Introductions to Hearing and Signal Analysis and a Glossary of Speech and Computer Terms*. 2nd Edn, Springer. http://books.google.co.za/books?id=cg7x5_-vB-AC&dq=speech+recognition&lr=&source=gbs_summary_s&cad=0

SHAIKH, R.H. and CHITODE, J.S. (2003): *Speech Synthesis for Devanagri Text*. Allied Publishers.

SHARWAR, B.A. and ATWELL, E. (2007): *Different Measurement metrics to evaluate a chatbot system. . Academic and Industrial Research in Dialog Technologies Workshop* ACM.

SHEIBER, S. (1994): *Lessons from a Restricted Turing Test*. *Communications of ACM* 37(6):70-78

SIEBEL SYSTEMS (2005): *Contact Center Analytics: Driving Greater Efficiency, Higher Revenue, and Increased Customer Satisfaction*, Siebel Systems.14.

SINGH, A. (2007): *An Intelligent User Interface Model for Contact Centre Operations*. Dissertation thesis. Computer Science, Nelson Mandela Metropolitan University. Port Elizabeth.

SUHM, B., BERS, J., MCCARTHY, D., FREEMAN, B., GETTY, D., GODFREY, K. and PETERSON, P. (2002): *Comparative Study of Speech in the Call Center: Natural Language Call Routing vs. Touch-Tone Menus*. *Chi Letters* 4(1)

TECHFAQ (2008): *What is Computer Telephony Integration (CTI)?* [online]. Available at http://www.tech-faq.com/computer-telephony-integration-cti.shtml [Accessed on 10th October 2008]

THE ASCENT GROUP (2006): *Improving IVR Self-Service; IVR Improvement Strategies* [online]. Available at www.ascentgroup.com [Accessed on 27th June 2007]

TOLLERVEY, N.H. (2006): *Custom Tags in AIML* [online]. Available at http://aimlbot.sourceforge.net/ [Accessed on 20th June 2007]

TREDOUX, C. and DURRHEIM, K. (2002): *Numbers, Hypothesis & Conclusions*. Cape Town, UCT Press.

TROCHIM, W.M.K. (2006): *Research Methods Knowledge Base* [online]. Available at http://www.socialresearchmethods.net/kb/index.php [Accessed on 20th November 2008]

TULLIS, T. and ALBERT, B. (2008): *Measuring The User Experience: Collecting, Analyzing and Presenting Usability Metrics*. 30 Corporate Drive, Suite 400, Burlington, MA, Elsevier, Morgan Kaufman Publishers.

TURING, A. (1950): *Can machines think? Computing Machinery and Intelligence*:433-460

VENTURA, J., VENTURA, M. and OLABE, J.C. (2005): Embodied conversational agents: developing usable agents [education instruction applications]. In *Proceedings of South East Conference*. 663- 669, IEEE

VOGTS, D. (1998): ***Design of a multimedia environment/Expert system shell for the identification problem*****. Dissertation thesis. Computer Science and Information Systems, University of Port Elizabeth. Port Elizabeth.**

VOXEO (2007): ***What is IVR?*** **[online]. Available at www.voxeo.com/automation/automation/ivr.jsp [Accessed on 27th September 2007]**

WALLACE, R.S. (2001): ***AIML Pattern Matching Simplified*** **[online]. Available at http://www.alicebot.org/documentation/matching.html [Accessed on 15th November 2007]**

WALLACE, R.S. (2005): ***Be your Own BotMaster: The step by step guide to creating, hosting and selling your own A.I. chatbot on pandorabots*****. Oakland, CA, ALICE A.I. Foundation, Inc.**

WELSH, C. (2006): ***Rule-Based/Expert Systems*** **[online]. Available at www.cse.lehigh.edu/~munoz/CSE335/classes/ExpertSystems.ppt [Accessed on 13th September 2008]**

WILCOX, B. (2008): ***Beyond AIML: Chatbots 102*** **[online]. Available at http://www.gamasutra.com/view/feature/3761/beyond_aiml_chatbots_102.php [Accessed on 20th August 2008]**

WINOGRAD, T. and FLORES, C.F. (1986): ***Understanding Computers and Cognition*****. Norwood New Jersey, Ablex.**

WOODWARD, N. (2007): ***Satisfaction Centre - not cost centre: focus on benefit, not just cost*****. Fujitsu. http://uk.fujitsu.com/POV/localData/pdf/satisfaction-centre.pdf**

WU, X., ICHIKAWA, T. and CERCONE, N. (1996): ***Knowledge-base Assisted Database Retrieval Systems*****. World Scientific.**

ZHOU, L., SHAIKH, M. and ZHANG, D. (2004): Natural Language Interface to Mobile Devices. In ***Proceedings of International Conference on Intelligent Information Processing (IIP2004)*****, Beijing, China. 543, SHI, Z. and HE, Q. (eds). Springer**

ZUE, V., COLE, R. and WARD, W. (2000): ***Speech Recognition*** **[online]. Available at http://cslu.cse.ogi.edu/HLTsurvey/ch1node4.html [Accessed on 20th September 2008]**

Appendix A

Natural Language Interface and Rule Based Expert System Pilot Study

---

PLEASE ANSWER THE QUESTIONS.

1. You have just finalised your dissertation and have to print it out for final submission. You click on the print icon and you notice that it does not print. You are fuming from the ears and decide to call on the NMMU Help desk. You also note that the printer is beeping at you and you notify the call taker (the call centre agent).
2. The same scenario as above but the dissertation prints. Upon further inspection of your dissertation you notice that there are funny colours in place of the right ones and some colours are missing. You have no idea what could have caused this and so you call the help desk.
3. The dissertation is printing and with about 20 pages to go the printer does not print anymore and a red light flashes and the printer is beeping at you. Your first instinct is to call the help desk in the fear of throwing the printer off the 15th floor of the main building.

Instructions

- Please fill out the questionnaire upon the completion of all tasks.

User feedback Questionnaire

Testing Rule Based and Pattern Matching Systems

Gender: ______________________

Age: ____

Occupation: _______________________________

ALICE (Pattern Matching)

Did you get all the information you wanted using the system? Please Elaborate.

____
____
____
____

Do you think the system understood what you asked?

____
____
____
____

How easy was it to obtain the information you wanted?

____
____
____
____

Was it difficult to reformulate your questions when you were invited to?

____
____
____
____

Do you think you would use this system again?

____
____
____
____

Overall, are you satisfied with the system?

____
____
____
____

Rule based System

Did you get all the information you wanted using the system? Please Elaborate.

____
____
____
____

Do you think the system understood what you asked?

____
____
____
____

How easy was it to obtain the information you wanted?

____
____
____
____

Do you think you would use this system again?

____
____
____
____

Overall, are you satisfied with the system?

____
____
____
____

Comparison of the two systems

Which system did you prefer to use?

____
____
____
____

Would you use the above mentioned system again?

____
____
____
____

Appendix B

Rule Based Natural Language Interface Pilot Study

PLEASE ANSWER THE QUESTIONS.

1. You have just finalised your dissertation and have to print it out for final submission. You click on the print icon and you notice that it does not print. You are fuming from the ears and decide to call on the NMMU Help desk. You also note that the printer is beeping at you and you notify the call taker (the call centre agent).
2. The same scenario as above but the dissertation prints. Upon further inspection of your dissertation you notice that there are funny colours in place of the right ones and some colours are missing. You have no idea what could have caused this and so you call the help desk.
3. The dissertation is printing and with about 20 pages to go the printer does not print anymore and a red light flashes and the printer is beeping at you. Your first instinct is to call the help desk in the fear of throwing the printer off the 15th floor of the main building.

Instructions

- Please fill out the questionnaire upon the completion of all tasks.

Nelson Mandela Metropolitan University

Department of Computer Science & Information Systems

This questionnaire is part of research towards a MSc in Computer Science and Information Systems

Pilot Study Questionnaire: PrintHal

PrintHal troubleshoots general printer problems (paper jam, toner problems) through the use of natural language. Problems are solved using ALICE pattern matching principles in combination with rule based logic.

	Section A: Biographical Details (mark with X where appropriate)					
1	Gender:	Male	Female			
2	Age:					
3	Occupation:					
4	Computer experience (years)	< 1	1 - 2	2 - 4	> 4	
5	Printer Troubleshooting Experience	< 1	1 - 2	2 - 4	> 4	N/A
6	Instant Messaging experience (years)	< 1	1 - 2	2 - 4	> 4	N/A
7	How often do you use instant messaging to communicate?	Never	About once a month	About once a week	A few times a week	Every day

Section B: Evaluation of System

Did you get all the information you wanted using the system? Please Elaborate.

Do you think the system understood what you asked? Please Elaborate.

How easy was it to obtain the information you wanted?
Was it difficult to reformulate your questions when you were invited to?
Do you think you would use this system again?
Overall, are you satisfied with the system?

Section C: General
Describe positive aspects of the system
Describe negative aspects of the system
Please provide any general comments or suggestions for improvement below

Appendix C

Final Evaluation

The Department of Computer Science & Information Systems

Usability Evaluation

Consent Form

Nelson Mandela Metropolitan University for tomorrow

You have been selected as a research participant for the evaluation of contact centre proofs of concepts. This evaluation is being conducted by Gopal Ravi Sankar (gopal.ravisankar@nmmu.ac.za). Please do not hesitate to ask if you have any questions about the evaluation. As a participant you have certain rights, which are listed below. You will be asked to perform various tasks with the relevant software. The purpose of this evaluation is to capture your interaction data as well as performance details. A secondary goal entails rating the usability of the software. We expect the session to last about 30 minutes. This data will be used only for research purposes and will not be distributed nor viewed by anyone not associated with this evaluation process. Your name will not be associated with any data that are collected during this evaluation session. There are no known risks associated with this evaluation. You will be asked to complete a feedback questionnaire, containing questions relevant to this evaluation.

Your rights as a participant are as follow:

1. You have the right to withdraw from the session at any time for any reason.
2. At the conclusion of the session, you may see your data if you so desire. If you decide to withdraw your data, please inform the evaluators immediately.
3. Finally, we greatly appreciate your time and effort for participating in this evaluation. Remember, you cannot fail any part of this session, and there are no right or wrong answers. Your signature below indicates that you have read this consent form in it’s entirely and that you voluntarily agree to participate.

Name & Surname:		Contact Tel. No.:	
Signature:		Date:	

Case Studies for ALICE

Please complete the following tasks, and complete the questionnaire that will be provided once all tasks have been completed.

Case Study One.

You have finalised an important document and want to print. You click the print button and notice that the printer is not doing anything. Upon further inspection you notice that the normal lights that are illuminated on the printer are not working anymore. Not knowing what to do you chat to your helpdesk.

1. **Start the Timer**.
2. Chat to the system, in order to solve the problem. You can start by saying hi.
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Case Study Two.

You have finalised an important document and want to print. You click the print button and the document starts printing. Half way through the printing cycle, the printer stops printing. Not knowing what to do, you decide to call the helpdesk.

1. **Start the Timer**.
2. Chat to the system, in order to solve the problem. You can start by saying hi
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Case Study Three.

You have finalised an important document and want to print. You click the print button and notice that the printer is not doing anything. The regular lights are on, but there is a red light flashing on the printer, you are not too sure as to what to do. You have therefore decided to chat to the helpdesk.

1. **Start the Timer**.
2. Chat to the system, in order to solve the problem. You can start by saying hi.
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Case Studies for IVR

Please complete the following tasks, and complete the questionnaire that will be provided once all tasks have been completed.

Case Study One.

You have finalised an important document and want to print. You click the print button and notice that the printer is not doing anything. Upon further inspection you notice that the normal lights that are illuminated on the printer are not working anymore. Not knowing what to do you chat to your helpdesk.

1. **Start the Timer**.
2. Follow the menus that are present in the system in order to solve your problem.
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Case Study Two.

You have finalised an important document and want to print. You click the print button and the document starts printing. Half way through the printing cycle, the printer stops printing. Not knowing what to do, you decide to call the helpdesk.

1. **Start the Timer**.
2. Follow the menus that are present in the system in order to solve your problem.
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Case Study Three.

You have finalised an important document and want to print. You click the print button and notice that the printer is not doing anything. The regular lights are on, but there is a red light flashing on the printer, you are not too sure as to what to do. You have therefore decided to chat to the helpdesk.

1. **Start the Timer**.
2. Follow the menus that are present in the system in order to solve your problem.
3. Once you think your problem is solved, **stop the timer** and record you time on the questionnaire.

Nelson Mandela Metropolitan University, Department of Computer Science and Information Systems

This questionnaire is part of research towards a M.Sc in Computer Science and Information Systems

Field Study Questionnaire

	Section A: Biographical Details (mark with X where appropriate)					
1	Gender:	Male	Female			
2	Age:					
3	Occupation:					
4	Computer experience (years)	< 1	1 - 2	3 - 4	> 4	
5	Printer experience (years)	< 1	1 - 2	3 - 4	> 4	N/A
6	Printer Troubleshooting experience (years)	< 1	1 - 2	3 - 4	> 4	N/A
5	Instant Messaging experience (years)	< 1	1 - 2	3 - 4	> 4	N/A
6	How often do you use instant messaging to communicate?	Never	About once a month	About once a week	A few times a week	Every day

	Section B1: ALICE					
1	Time Taken To complete task:					
2	Did you get all the information you wanted using the system?	Never				Always
		1	2	3	4	5
3	Do you think the system understood what you asked?	Never				Always
		1	2	3	4	5
4	How easy was it to obtain the information you wanted?	Very Easy				Very Difficult
		1	2	3	4	5
5	How difficult was it to reformulate your questions when you were invited to?	Very Easy				Very Difficult
		1	2	3	4	5
6	Would you use this system again?	Definitely Yes				Definitely No

		1	2	3	4	5
	Section B2: IVR					
1	Time Taken To complete task:					
2	Did you get all the information you wanted using the system?	Never				Always
		1	2	3	4	5
3	Do you think the system understood what you asked?	Never				Always
		1	2	3	4	5
4	How easy was it to obtain the information you wanted?	Very Easy				Very Difficult
		1	2	3	4	5
5	Would you use this system again?	Definitely Yes				Definitely No
		1	2	3	4	5
	Section B3: Hybrid					
1	Time Taken To complete task:					
2	Did you get all the information you wanted using the system?	Never				Always
		1	2	3	4	5
3	Do you think the system understood what you asked?	Never				Always
		1	2	3	4	5
4	How easy was it to obtain the information you wanted?	Very Easy				Very Difficult
		1	2	3	4	5
5	How difficult was it to reformulate your questions when you were invited to?	Very Easy				Very Difficult
		1	2	3	4	5
6	Would you use this system again?	Definitely Yes				Definitely No
		1	2	3	4	5
	Section B4: Preference	**Rank (1=Best) (3=Worst)**				
	ALICE					
	IVR					
	Hybrid					

Section C: General		
Describe positive aspects of the systems (if any):		
ALICE	IVR	Hybrid
Describe negative aspects of the systems (if any):		
ALICE	IVR	Hybrid
Please provide any general comments or suggestions for improvement to the systems:		

Appendix D

Biographical Profile of Test Participants

<u>Biographical Data</u>

Age (Years)			**Gender**		**Computer Experience(Years)**				**Printer Experience(Years)**					**Printer Troubleshooting Experience**				
20-30	*30-40*	*>40*	*Male*	*Female*	*<1*	*1-2*	*3-4*	*>4*	*<1*	*1-2*	*3-4*	*>4*	*N/A*	*<1*	*1-2*	*3-4*	*>4*	*N/A*
14	3	3	12	8	1	1	0	18	2	0	3	15	0	5	3	1	8	3

<u>Biographical Data (Continued)</u>

Instant Messaging Experience (Years)					**How often do you use instant messaging to communicate**				
<1	*1-2*	*3-4*	*>4*	*N/A*	*Never*	*About Once a Month*	*About once a week*	*A few time a week*	*Everyday*
1	3	4	11	1	2	4	0	5	9

Table D.1 Biographical Data of Test Participants (N=20)

Appendix E

Quantitative Results

Variable	Descriptive Statistics							
	Valid N	Mean	Median	Mode	Frequency of Mode	Minimum	Maximum	Std.Dev.
IVR_Q1	20	3.700000	4.000000	4.000000	7	1.000000	5.000000	1.080935
IVR_Q2	20	3.550000	4.000000	4.000000	7	1.000000	5.000000	1.234376
IVR_Q3r	20	3.200000	3.000000	Multiple	6	1.000000	5.000000	1.196486
IVR_Q4r	20	3.000000	3.000000	3.000000	7	1.000000	5.000000	1.337712

Table E.1 IVR User Likert Scale Ratings

Variable	Descriptive Statistics							
	Valid N	Mean	Median	Mode	Frequency of Mode	Minimum	Maximum	Std.Dev.
ALICE_Q1	20	3.050000	2.500000	2.000000	9	1.000000	5.000000	1.316894
ALICE_Q2	20	2.500000	2.000000	2.000000	12	1.000000	5.000000	1.000000
ALICE_Q3r	20	2.200000	2.000000	Multiple	6	1.000000	4.000000	1.005249
ALICE_Q4r	20	1.950000	2.000000	2.000000	8	1.000000	4.000000	0.887041
ALICE_Q5r	20	1.750000	2.000000	2.000000	9	1.000000	3.000000	0.716350

Table E.2 ALICE User Likert Scale Ratings

Variable	Descriptive Statistics							
	Valid N	Mean	Median	Mode	Frequency of Mode	Minimum	Maximum	Std.Dev.
Hybrid_Q1	20	4.400000	4.000000	4.000000	12	4.000000	5.000000	0.502625
Hybrid_Q2	20	4.200000	4.000000	5.000000	9	2.000000	5.000000	0.951453
Hybrid_Q3r	20	4.100000	4.000000	5.000000	9	2.000000	5.000000	1.020836
Hybrid_Q4r	20	4.200000	4.000000	Multiple	8	3.000000	5.000000	0.767772
Hybrid_Q5r	20	4.350000	5.000000	5.000000	11	2.000000	5.000000	0.933302

Table E.3 Hybrid User Likert Scale Ratings

Category	Frequency table: IVR_rank Count	Cumulative Count	Percent	Cumulative Percent
1	3	3	15.00000	15.0000
2	12	15	60.00000	75.0000
3	5	20	25.00000	100.0000
Missing	0	20	0.00000	100.0000

Table E.4 IVR User Ranking

Category	Frequency table: ALICE_rank Count	Cumulative Count	Percent	Cumulative Percent
2	5	5	25.00000	25.0000
3	15	20	75.00000	100.0000
Missing	0	20	0.00000	100.0000

Table E.5 ALICE User Ranking

Category	Frequency table: Hybrid_rank Count	Cumulative Count	Percent	Cumulative Percent
1	17	17	85.00000	85.0000
2	3	20	15.00000	100.0000
Missing	0	20	0.00000	100.0000

Table E.6 Hybrid User Ranking

Variable	Descriptive Statistics Valid N	Mean	Mode	Frequency of Mode	Minimum	Maximum	Std.Dev.
IVR_rank	20	2.100000	2.000000	12	1.000000	3.000000	0.640723
ALICE_rank	20	2.750000	3.000000	15	2.000000	3.000000	0.444262
Hybrid_rank	20	1.150000	1.000000	17	1.000000	2.000000	0.366348

Table E.7 Comparison of Three Prototype Rankings

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000 Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR1	54.35000	24.75836						
Hybrid1	66.65000	59.56003	20	-12.3000	66.55358	-0.826511	19	0.418773

Table E.8 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
IVR1 & Hybrid1	20	102.5000	0.093332	0.925640

Table E.9 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
ALICE1	86.50000	65.51778						
Hybrid1	66.65000	59.56003	20	19.85000	31.91151	2.781814	19	0.011885

Table E.10 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
ALICE1 & Hybrid1	20	37.00000	2.334047	0.019594

Table E.11 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR1	54.35000	24.75836						
ALICE1	86.50000	65.51778	20	-32.1500	71.45501	-2.01216	19	0.058599

Table E.12 *T-test comparing time taken to complete each task on each prototype*

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
IVR1 & ALICE1	20	63.00000	1.567972	0.116889

Table E.13 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR2	44.45000	25.42580						
Hybrid2	78.10000	61.47306	20	-33.6500	63.94675	-2.35332	19	0.029528

Table E.14 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
IVR2 & Hybrid2	20	29.00000	2.837283	0.004550

Table E.15 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
ALICE2	83.46667	59.40042						
Hybrid2	61.86667	29.22051	15	21.60000	72.01865	1.161594	14	0.264822

Table E.16 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
ALICE2 & Hybrid2	15	35.50000	1.391506	0.164073

Table E.17 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR2	43.46667	26.30825						
ALICE2	83.46667	59.40042	15	-40.0000	70.75713	-2.18945	14	0.046000

Table E.18 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
IVR2 & ALICE2	15	13.00000	2.669421	0.007599

Table E.19 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR3	44.00000	15.73130						
Hybrid3	97.85000	33.09679	20	-53.8500	39.06711	-6.16438	19	0.000006

Table E.20 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
IVR3 & Hybrid3	20	5.500000	3.714601	0.000204

Table E.21 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000							
	Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
ALICE3	102.9286	52.44406						
Hybrid3	103.2857	32.68834	14	-0.357143	62.26199	-0.021463	13	0.983203

Table E.22 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000			
	Valid N	T	Z	p-level
ALICE3 & Hybrid3	14	25.00000	1.432656	0.151957

Table E.23 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	T-test for Dependent Samples (Statistics - for Dissertation.sta) Marked differences are significant at p < .05000 Mean	Std.Dv.	N	Diff.	Std.Dv. Diff.	t	df	p
IVR3	40.2857	10.78766						
ALICE3	102.9286	52.44406	14	-62.6429	54.58319	-4.29414	13	0.000873

Table E.24 T-test comparing time taken to complete each task on each prototype

Pair of Variables	Wilcoxon Matched Pairs Test (Statistics - for Dissertation.sta) Marked tests are significant at p <.05000 Valid N	T	Z	p-level
IVR3 & ALICE3	14	0.00	3.295765	0.000982

Table E.25 Wilcoxon matched pairs test comparing time taken to complete each task on each prototype

Variable	Descriptive Statistics Valid N	Mean	Minimum	Maximum	Std.Dev.
IVR_effect	20	3.6	1.0	5.0	1.1
IVR_effic	20	3.2	1.0	5.0	1.2
IVR_satisf	20	3.0	1.0	5.0	1.3
ALICE_effect	20	2.8	1.0	5.0	1.1
ALICE_effic	20	2.1	1.0	3.0	0.7
ALICE_satisf	20	1.8	1.0	3.0	0.7
Hybrid_effect	20	4.3	3.0	5.0	0.6
Hybrid_effic	20	4.2	2.5	5.0	0.8
Hybrid_satisf	20	4.3	2.0	5.0	0.9

Table E.26 Comparison of Usability Metrics across three prototypes

Variable	Spearman Rank Order Correlations (Statistics - for Dissertation.sta) MD pairwise deleted Marked correlations are significant at p <.05000 IVR1	IVR2	IVR3	IVR_effect	IVR_effic	IVR_satisf
IVR1	1.00	0.12	0.22	-0.21	-0.04	-0.17
IVR2	0.12	1.00	0.42	-0.04	0.38	0.19
IVR3	0.22	0.42	1.00	-0.16	0.09	0.01
IVR_effect	-0.21	-0.04	-0.16	1.00	0.32	0.54
IVR_effic	-0.04	0.38	0.09	0.32	1.00	0.73
IVR_satisf	-0.17	0.19	0.01	0.54	0.73	1.00

Table E.27 Time taken VS Likert scale rating for IVR

Variable	Spearman Rank Order Correlations (Statistics - for Dissertation.sta) MD pairwise deleted Marked correlations are significant at p <.05000 ALICE1	ALICE2	ALICE3	ALICE_effect	ALICE_effic	ALICE_satisf
ALICE1	1.00	0.26	0.48	-0.40	-0.22	-0.13
ALICE2	0.26	1.00	0.81	-0.30	-0.15	-0.17
ALICE3	0.48	0.81	1.00	-0.31	-0.07	-0.14
ALICE_effect	-0.40	-0.30	-0.31	1.00	0.60	0.36
ALICE_effic	-0.22	-0.15	-0.07	0.60	1.00	0.60
ALICE_satisf	-0.13	-0.17	-0.14	0.36	0.60	1.00

Table E.28 Time taken VS Likert scale rating for NLI

Variable	Spearman Rank Order Correlations (Statistics - for Dissertation.sta) MD pairwise deleted Marked correlations are significant at p <.05000 Hybrid1	Hybrid2	Hybrid3	Hybrid_effect	Hybrid_effic	Hybrid_satisf
Hybrid1	1.00	0.16	-0.24	-0.42	-0.27	-0.46
Hybrid2	0.16	1.00	0.20	-0.07	-0.16	-0.07
Hybrid3	-0.24	0.20	1.00	0.60	0.08	0.32
Hybrid_effect	-0.42	-0.07	0.60	1.00	0.24	0.46
Hybrid_effic	-0.27	-0.16	0.08	0.24	1.00	0.36
Hybrid_satisf	-0.46	-0.07	0.32	0.46	0.36	1.00

Table E.29 Time taken VS Likert scale rating for Hybrid

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	643.5375	1	643.5375	205.3412	0.000000
Error	59.5458	19	3.1340		
IVR	4.0750	2	2.0375	3.0857	0.057308
Error	25.0917	38	0.6603		

Table E.30 Anova test for IVR

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	290.4000	1	290.4000	208.7364	0.000000
Error	26.4333	19	1.3912		
ALICE	10.9750	2	5.4875	14.6935	0.000019
Error	14.1917	38	0.3735		

Table E.31 Anova test for NLI

Cell No.	Tukey HSD test; variable DV_1 Approximate Probabilities for Post Hoc Tests Error: Within MSE = .37346, df = 38.000 ALICE	{1} 2.7750	{2} 2.0750	{3} 1.7500
1	ALICE_effect		0.002489	0.000135
2	ALICE_effic	0.002489		0.225349
3	ALICE_satisf	0.000135	0.225349	

Table E.32 Tukey HSD test for NLI

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	1092.267	1	1092.267	1051.676	0.000000
Error	19.733	19	1.039		
HYBRID	0.433	2	0.217	0.497	0.612263
Error	16.567	38	0.436		

Table E.33 Anova test for Hybrid

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	763.2667	1	763.2667	432.0377	0.000000
Error	33.5667	19	1.7667		
EFFECT	23.3583	2	11.6792	23.5964	0.000000
Error	18.8083	38	0.4950		

Table E.34 Anova test for effectivness

Cell No.	Tukey HSD test; variable DV_1 Approximate Probabilities for Post Hoc Tests Error: Within MSE = .49496, df = 38.000 EFFECT	{1} 3.6250	{2} 2.7750	{3} 4.3000
1	IVR_effect		0.001460	0.011892
2	ALICE_effect	0.001460		0.000124
3	Hybrid_effect	0.011892	0.000124	

Table E.35 Tukey HSD test for effectiveness

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	592.2042	1	592.2042	739.6469	0.000000
Error	15.2125	19	0.8007		
EFFIC	43.1583	2	21.5792	24.3507	0.000000
Error	33.6750	38	0.8862		

Table E.36 Anova test for effciency

Cell No.	Tukey HSD test; variable DV_1 Approximate Probabilities for Post Hoc Tests Error: Within MSE = .88618, df = 38.000 EFFIC	{1} 3.2000	{2} 2.0750	{3} 4.1500
1	IVR_effic		0.001631	0.007921
2	ALICE_effic	0.001631		0.000124
3	Hybrid_effic	0.007921	0.000124	

Table E.37 Tukey HSD test for effciency

Effect	Repeated Measures Analysis of Variance Sigma-restricted parameterization Effective hypothesis decomposition SS	Degr. of Freedom	MS	F	p
Intercept	552.0667	1	552.0667	718.4429	0.000000
Error	14.6000	19	0.7684		
SATISF	67.6333	2	33.8167	28.1189	0.000000
Error	45.7000	38	1.2026		

Table E.38 Anova test for satisfaction

Cell No.	Tukey HSD test; variable DV_1 Approximate Probabilities for Post Hoc Tests Error: Within MSE = 1.2026, df = 38.000 SATISF	{1} 3.0000	{2} 1.7500	{3} 4.3500
1	IVR_satisf		0.002611	0.001206
2	ALICE_satisf	0.002611		0.000124
3	Hybrid_satisf	0.001206	0.000124	

Table E.39 Tukey HSD test for satisfaction

Appendix F

Code Snippets

```
namespace AIMLbot.Normalize
{
    /// <summary>
    /// Strips any illegal characters found within the input string. Illegal
characters are referenced from
    /// the bot's Strippers regex that is defined in the setup XML file.
    /// </summary>
    public class StripIllegalCharacters : AIMLbot.Utils.TextTransformer
    {
        public StripIllegalCharacters(AIMLbot.Bot bot, string inputString) :
base(bot, inputString)
        { }

        public StripIllegalCharacters(AIMLbot.Bot bot)
            : base(bot)
        {

        }

        protected override string ProcessChange()
        {
            return this.bot.Strippers.Replace(this.inputString, " ");
        }
```

Normalization in the ALICE System

```
  <item name="aimldirectory" value="aiml"/>

  <item name="configdirectory" value="config"/>

  <item name="logdirectory" value="logs"/>

  <item name="splittersfile" value="Splitters.xml"/>

  <item name="defaultpredicates" value="DefaultPredicates.xml"/>

  <item name="substitutionsfile" value="Substitutions.xml"/>

  <item name="maxlogbuffersize" value="64"/>

  <item name="notacceptinguserinputmessage" value="This bot is currently set
to not accept user input."/>

  <item name="stripperregex" value="[^0-9a-zA-Z(){}]"/>
```

Regular Expression Stripper Value

Creating Custom Tags

The first step is to create a .Net Class library and once the shell of the class is created the AIMLBOT.DLL must be referenced. Each custom tag has its own class that is utilised and has the same name as that of the tag in the AIML. The most important thing to note is that the class must have a [Custom Tag] attribute associated with it (Figure.5.19).and must inherit from the AIMLTagHandler Class (Tollervey 2006) (Figure 5.18). The AIMLTagHandler class is responsible for getting and setting AIML predicates at runtime.

The class constructor must assign a value for the inputString attribute and this is usually assigned the same name as the tag name to be utilised. The standard AIML tags are handled by classes which inherit from the AIMLTagHandler class and they make use of inputString but a custom tag does not have to. The new functionality can now be added to the OverridenProcessChange method. The output of this method is a string which contains the raw output.

The last step involves loading the DLL created into your ALICE chatbot utilising the loadCustomTagHandlers(String pathToDLL) mehod. As the bot processes the template part of your AIML category it will attempt to find in the hash table an instance of a handler class with the same name as the custom tag and use it to process the XML node in question.

See Sample Code on Next Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;
using AIMLbot.Utils;
using System.IO;
using System.Collections;
using System.Data;
using System.ComponentModel;

namespace RulesCustomTag
{
    [CustomTag]
    public class rule : AIMLTagHandler
    {
        AIMLGUI.aimlForm forms;
        static rule instance;
        public rule()
        {

            this.inputString = "rule";
            instance = this;
        }

        public static rule getInstance()
        {
            if (instance == null)
            {
                new rule();
            }
            return instance;
        }

        public AIMLGUI.aimlForm getForm()
        {
            if (forms == null)
            {
                forms = new AIMLGUI.aimlForm();
            }
            return forms;
        }

        protected override string ProcessChange()
        {
            if (this.templateNode.Name.ToLower() == "rule")
            {
                sendlist(this.templateNode.InnerText);
                return string.Empty;
            }
            return "Not Working";
        }
        public void sendlist(String rule)
        {
           getForm().receivelist(rule);
        }
    }
```