

2007

On the resolution of system generated user interruption in context-aware systems

Ashish Godbole
University of Dayton

Follow this and additional works at: https://ecommons.udayton.edu/graduate_theses

Recommended Citation

Godbole, Ashish, "On the resolution of system generated user interruption in context-aware systems" (2007). *Graduate Theses and Dissertations*. 2926.
https://ecommons.udayton.edu/graduate_theses/2926

This Dissertation is brought to you for free and open access by the Theses and Dissertations at eCommons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of eCommons. For more information, please contact mschlange1@udayton.edu, ecommons@udayton.edu.

116519

ON THE RESOLUTION OF SYSTEM GENERATED USER INTERRUPTION IN
CONTEXT-AWARE SYSTEMS

DISSERTATION

Submitted to the

The School of Engineering

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree

Doctor of Philosophy in Electrical Engineering

by

Ashish Godbole

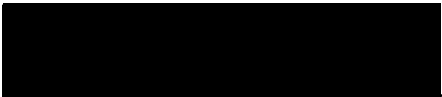
UNIVERSITY OF DAYTON

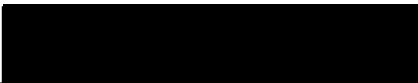
Dayton, Ohio

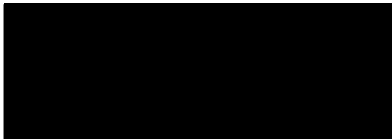
May, 2007

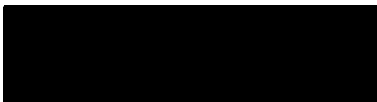
ON THE RESOLUTION OF SYSTEM GENERATED USER INTERRUPTION IN
CONTEXT-AWARE SYSTEMS

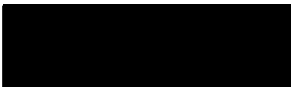
APPROVED BY:

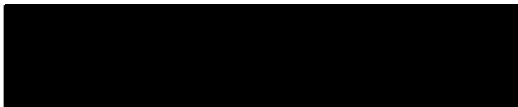

Malcolm W. Daniels, Ph.D.
Advisory Committee Chairman
Associate Professor, Electrical and
Computer Engineering Department


John S. Loomis, Ph.D.
Committee Member
Associate Professor, Electrical and
Computer Engineering Department


Gerald J. Shaughnessy, M.S.
Committee Member
Associate Professor,
Department of Mathematics


Donald L. Moon, Ph.D.
Committee Member
Professor, Electrical and
Computer Engineering Department


Donald L. Moon, Ph.D.
Associate Dean
Graduate Engineering Programs &
Research, School of Engineering


Joseph E. Saliba, Ph.D., P.E.
Dean, School of Engineering

ABSTRACT

ON THE RESOLUTION OF SYSTEM GENERATED USER INTERRUPTION IN CONTEXT-AWARE SYSTEMS

Name: Godbole, Ashish
University of Dayton

Advisor: Dr. Malcolm W. Daniels

Over the past few decades, electronic devices and computing systems have become highly flexible, responsive and user friendly. While such properties have provided many advantages, they have also led to an increase in the amount, type and frequency of interruptions caused to user activity. These interruptions cannot be eliminated since they are necessary for activities such as notifications, interaction, information exchange, and so on. Their effects however are a cause for concern, and may range from mildly annoying to highly catastrophic in nature. This research specifically focuses on handling system-generated interruptions during human-system interaction in context-aware systems. The approach is to channel interruptions not by removing their sources, but by making them timely and appropriate to the given situation. Currently there are few solutions for interruption handling, but a majority of them are static, rule based or application specific. This research instead employs machine learning techniques and models that determine how to interrupt the user in ever changing situations. The goal of this approach is that whenever a system is about to interrupt a user, it employs its learned

models to modify the interruptions and make them less intrusive. To make this process more accurate, information is elicited from users about their activities, surroundings and preferences to interruptions. Two popular and diverse scenarios namely cell phones and office secretaries are developed to test this approach. The main advantage of this approach is that it covers a broader range of situations and scenarios than static, rule based or user training based approaches. The results show that interruptions are handled effectively and at the same time, user preferences are also taken into account. Furthermore, the ability to allow users to directly modify system properties in real time makes the approach applicable to changing situations. For the scenarios considered, the approach is not overly limited by technological constraints and current tools will make it possible to a large degree to implement this approach in actual systems.

ACKNOWLEDGMENTS

I would like to thank the following faculty, staff, and students for supporting me throughout the various endeavors that I undertook at the University of Dayton.

Dr. Moon, Dr. Daniels, and Dr. Banerjee for providing me with opportunities to teach, conduct research, and receive the highest level of education at UD. Thank you for helping me keep my research on track, and my mind focused despite many obstacles. Mr. Shaughnessy, and Dr. Loomis for providing valuable guidance on research activities. Dr. Moroney, and Dr. Swierenga for introducing me to the world of Human Computer Interaction. Dr. Smari for supporting my studies and research for so many years. I hope to apply your tireless work ethic in my workplace and life. Dr. Berney, and Dr. Smith for providing me excellent opportunities and much needed advice in the field of teaching.

Marilyn Knisley, and Loretta Christon, who are probably the two most hardworking staff members at UD. Thank you for your help and especially your immeasurable patience.

Rebecca, for helping me to compile literally millions of lines of data for my research, and for introducing me to the absolutely incomparable sport of skydiving. Kim, for being such a great research buddy, and sharing so many experiences with me while at UD.

Bonnie Simmons, an amazing person without whom I would be neck deep in library fines.

On a personal note, this work would not have been possible without the patience, support, blessings, and most importantly love of my family and friends. Anna, Ajji, Neelambari, Aai, and Pappa, this is for you.

TABLE OF CONTENTS

	Page
Abstract.....	iii
Acknowledgments.....	v
Table of Contents.....	vi
List of Figures.....	ix
List of Tables.....	xiii
CHAPTERS	
1. INTRODUCTION	1
1.1. Scope of the Dissertation	3
1.2. Dissertation Outline	4
1.3. Dissertation Contributions	5
2. BACKGROUND AND FUNDAMENTAL RESEARCH	8
2.1. Human Centeredness	9
2.2. Context-Aware Systems.....	13
2.3. User Interruption.....	19
3. PROBLEM DEFINITION: SYSTEM GENERATED USER INTERRUPTION.....	25
3.1. Problem Definition.....	25
3.2. Reasons for Selecting the Problem	26
4. EXISTING APPROACHES AND PROPOSED SOLUTION	30
4.1. Existing Solutions	30
4.1.1. User Driven Approaches.....	31
4.1.2. System Driven Approaches.....	34
4.1.2.1. Enhancing System Design	35

4.1.2.2. Enhancing Interruption Design	37
4.2. Proposed Solution	39
4.2.1. Pre-design Information Elicitation.....	42
4.2.2. Decision Tree-based Learning	45
4.2.3. Post-design Preference Elicitation	49
5. APPLICATION DOMAINS AND SCENARIOS	52
5.1. Desktop PCs.....	56
5.2. Instant Messaging	57
5.3. Cell Phone Scenario.....	58
5.4. Virtual Secretary Scenario	61
6. INITIAL DESIGN AND ANALYSIS.....	65
6.1. Design Methodology.....	66
6.2. Information Elicitation.....	69
6.2.1. Field Studies.....	70
6.2.2. Experience Sampling	74
6.2.3. User Surveys	80
6.3. Information Analysis	82
6.4. Scenario Development.....	102
6.4.1. Cell Phone.....	103
6.4.2. Virtual Secretary	107
6.5. Training Data Collection.....	110
7. MODELING AND SIMULATION OF SCENARIOS	114
7.1. Architecture for Simulation	114
7.2. Enabling Technologies.....	118
7.2.1. Java SDK and J2ME	119
7.2.2. WEKA.....	120
7.2.3. Modeling Using J48 Decision Trees.....	123
7.2.4. Palm OS and IBM WEME.....	127
7.3. Experimentation.....	129
8. EVALUATION AND VALIDATION OF SIMULATOR.....	136
8.1. Evaluation of Simulator	137
8.1.1. Training Data Analysis	137
8.1.2. Evaluation of Attributes.....	144
8.2. User Validation	149

9. IMPLEMENTATION ISSUES	161
9.1. General Implementation Issues.....	161
9.2. System Design Technologies.....	163
9.2.1. Sensors	163
9.2.2. Network Infrastructure.....	168
9.2.3. Computing Systems and Devices.....	170
9.2.4. User Environments.....	172
9.3. Attribute Alternatives.....	175
10. CONCLUSIONS AND FUTURE WORK	178
10.1. Summary and Conclusions	178
10.2. Main Contributions	180
10.3. Future Work.....	183
10.3.1. Prototyping.....	183
10.3.2. Automating Information Elicitation.....	185
10.3.3. Alternative Approaches	189
10.4. Possible Directions.....	194
APPENDICES	198
BIBLIOGRAPHY	213

LIST OF FIGURES

Figure No.	Page
1. Background Discussion and Fundamental Work in Three Areas Relevant to this Research	8
2. Three Perspectives for Human Centric System Design.....	11
3. Preliminary Framework for Human-Centric System Design.....	12
4. Context Classification Taxonomy.....	16
5. A Generic Process of Context-Aware Adaptation	17
6. The Basic Fundamentals of System Generated User Interruption.....	21
7. Taxonomy of Solutions for Handling System-generated Interruption	31
8. Proposed Approach Employing Pre and Post Design Elicitation for Learning Accurate Interruption-aware Models.....	41
9. Cell Phone Scenario with Possible Sensors and User Contexts	59
10. Virtual Secretary Scenario with Sensors, User Activities and Locations.....	62
11. Methodology for the Design of Interruption-Aware Systems	66
12. Incorporating Elicited Information to Build Interruption-Aware Models	69
13. PDA-Based Experience Sampling for Generating Interruption-Aware Learning Models	77
14. Sample PDA Based Experience Sampling Form.....	77
15. Sample Questionnaire for the Cell Phone Scenario	83
16. Percentage Distribution of Notification Modes on Respondents' Cell Phones	88

17. Percentage Distribution of Default Notification Modes Used by Cell Phone Owners	88
18. Percentage Distribution of Users Changing their Default Notification Modes for Noise Sensitive and at Home Locations.....	89
19. Percentage Distribution of Users Changing their Default Notification Modes when in Work Related Locations.....	90
20. Preferences for Situation Awareness Functionality, and Features Indicating Caller's Purpose in Cell Phones	90
21. Preference of Notification Modes in High-Noise and Noise Sensitive Situations.....	91
22. Frequency of Combined Active and Passive Notification Modes in Case of Emergencies Irrespective of Location and Situation.....	92
23. Deciding Whether to Answer Based on Four Choices in Home Situations, Work Related Situations, Public Places, and in Vehicles.....	93
24. Percentage of Work Related Calls Picked Up Based on User Situation.....	94
25. Number of Respondents Preferring Given Notification Modes for Various Situations in Case of Unknown Calls	94
26. Notification Mode Preferences for Family or Friend Emergencies for Different Situations.....	95
27. Percentage of Users Answering Other Users' Phones.....	95
28. Preferred Phone Actions when User Forgets Phone at Given Locations.....	96
29. Preferences of Notification Modes when Phones Have Been Forgotten at the Given Places.....	96
30. Percentage of Respondents Preferring Different Notification Modes when Performing Three Activities with their Cell Phones.....	97
31. Phone Locations when User is in Home Based Situations, Work Related Situations, Public Places, and in Vehicles	98
32. List of Possible Attributes and their Values for the Cell Phone Scenario	104

33.	List of Possible Attributes and their Values for the Virtual Secretary Scenario	108
34.	Dual Model Approach for the Virtual Secretary Scenario.....	109
35.	Notification Mode Preferences in Case of Emergency Calls.....	111
36.	Notification Mode Preferences when Phone is Not with the Users.....	112
37.	Generic Architecture of Context-Aware System	115
38.	Architecture of the Java-based Interruption Handling Simulator	116
39.	A Sample J48 Decision Tree.....	123
40.	GUI Output of Decision Tree Model Results	126
41.	Screen 1 Displaying Basic Options for Data Creation, Conversion and Selection.....	129
42.	Screen 2 Displaying Options to Create, Select and Test the Decision Tree Models of the Two Scenarios	130
43.	Screen Showing Default Simulator Screen for Displaying Results of the Interruption Handling Process	131
44.	Table Displaying Instance Information Along with the Response Class	132
45.	Simulator Screen Displaying Change in Graphics (and Audio) for Each Newly Tested Instance for the Virtual Secretary Scenario.....	133
46.	Simulator Screen Displaying Results for Each of the Five Sample Test Instances.....	134
47.	Variation of Dataset Size for Both Manual and Randomly Sampled Experiments	139
48.	Time Taken to Build Decision Tree Model for Manually Sampled and Randomly Sampled Training Datasets.....	140
49.	Classification Accuracy for Manual and Randomly Sampled Datasets	141
50.	Accuracy of Model for Varying Training Dataset Sizes	142
51.	Mean Absolute Error Rate for Virtual Secretary Scenario	142

52.	Use Case Comparisons for Virtual Secretary Scenario	153
53.	Use Case Comparisons for Virtual Secretary Scenario Continued.....	154
54.	Use Case Comparisons for Cell Phone Scenario	156
55.	Use Case Comparisons for Cell Phone Scenario Continued	157
56.	Typical Sensor Connectivity in an Interruption Aware System	167
57.	An Example Pervasive and Distributed Environment with Mixed Network Infrastructure.....	168
58.	Initial Prototype Architecture of Interruption Aware Cell Phone Simulator	184
59.	Initial Prototype Architecture of Interruption Aware Virtual Secretary Simulator.....	184

LIST OF TABLES

Table No.	Page
1. Sensor Attributes, Values and Responses for Two Sample Situations.....	47
2. Sample Use Cases and Possible Sensor Attributes for the PC Scenario.....	56
3. Sample Use Cases and Possible Sensor Attributes for the Instant Messaging Scenario	57
4. Sample Situations, Context Attributes and Interruptability Responses in Cell Phone Scenario.....	60
5. Sample Situations, Context Attributes and Interruptability Responses in Virtual Secretary Scenario	63
6. Information Obtained from Direct Observations and Informal Interviews of Departmental Secretaries in University Professors' Offices	72
7. Information Obtained from Informal Interviews of Cell Phone Users	73
8. Summary of PDA Based Experience Sampling.....	78
9. List of Notification Modes and Different Types of Levels for Mobile Devices.....	87
10. Common Activities and Places of Respondents Along with Data About Most Probable Phone Locations and Preferred Notification Modes	99
11. Common Noise-Sensitive Activities and Places of Respondents with Data About Most Probable Phone Locations and Preferred Notification Modes.....	99
12. List of Activities Performed by a Student Assistant or Department Secretary	100
13. Average Number of Different Types of Visitors	101

14.	Secretary Responses to Incoming Visitors	102
15.	List of Sensor Attributes and their Values for the Cell Phone Scenario.....	105
16.	List of Sensor Attributes and their Values for the Virtual Secretary Scenario.....	108
17.	Sample Set of Instances Before Being Classified for Training Data Creation.....	111
18.	Sample Training Dataset Created by Integrating Survey Information with Sample Instances.....	113
19.	Summary of Tools Employed for Developing the Simulation Architecture.....	117
20.	Training Dataset for Decision Tree in Figure 45	124
21.	Results of Experimentation on Interruption Handling Scenarios	134
22.	Varying the Training Dataset Size for Comparison of Model Performance	138
23.	Properties of the Training Datasets for the Virtual Secretary Scenario.....	139
24.	Number of Incorrectly Classified Instances for Various Dataset Sizes.....	143
25.	Attribute and Attribute Subset Evaluation Algorithms in WEKA.....	145
26.	Attribute Search and Ranking Algorithms in WEKA.....	146
27.	Ranked Results of Two Attribute Subset Evaluation and Selection Techniques	147
28.	Ranked Results of Two Individual Attribute Evaluation and Ranking Techniques	147
29.	Comparison of Classification Percentage for Original and Reduced Training Datasets for the Virtual Secretary Scenario	149
30.	Use Cases for Validating the Cell Phone Scenario	152
31.	Use Cases for Validating the Virtual Secretary Scenario	152

32.	Possible List of Sensors for Interruption Aware System Design.....	164
33.	Different Levels in Pre-design Information Elicitation	186
34.	Frequency Table for Storing Decision Tree Instance Information	193

CHAPTER 1

INTRODUCTION

Electronic devices, computing systems and their applications today are no longer limited to experienced users. With the growing variety of deployment scenarios, environments, and end users, designers are incorporating features that provide user friendliness and human centric capabilities to their designs. In other words, systems being developed today not only provide a wide range of services, they also allow users with different levels of expertise, cultures, and physical abilities to perform a broad range of activities in diverse physical environments. In designing such systems, areas from both human factors and engineering disciplines are being studied in parallel because of the complex inter-relationship between the system, the user and her environment. For example, human factors areas such as usability, ergonomics, psychology, linguistics, and social sciences help designers in understanding how users interact with devices and systems in their everyday environments. They aid in understanding users' behavior, characteristics, group dynamics and other issues that are relevant when systems are deployed over cultural, geographic, or organizational divisions. They also help cater to individual user needs and preferences in addition to their constraints and limitations. Engineering and Computer Science disciplines on the other hand such as Artificial

Intelligence, Pervasive Computing, Human Computer Interaction, and Context Aware Systems allow designers to incorporate a variety of fascinating features in systems. For example, artificial intelligence and context aware design make systems intelligent, responsive, emotive and aware. Areas such as pervasive and ubiquitous computing make systems transparent to the user and fit seamlessly in her environment.

Although these areas offer great promise and over the years have yielded a considerable number of commercial and scientific applications, they are not without their drawbacks which may range from minor faults and annoyances to users, to catastrophic and irrecoverable errors. Perhaps the most popular discussion on such drawbacks was provided by the late Michael Dertouzos [Dertouzos] who suggested that in order for systems to be truly user friendly and human centric, the approach to current research on intelligent systems should focus more on augmentation and assistance than on replication. His argument centered on the fact that systems that assist users by enhancing or complementing the communications, resources and services structure, are more likely to be effective than systems that attempt to replicate user behavior, activity or cognitive abilities. For example, intelligent animated assistants (such as the early Microsoft Office assistant 'Clippy') do not always provide the right assistance to the user, but often cause hindrance to her activities by 'popping up' at random and distracting her from her work. Engineering and Computer Science literature is replete with designs that offer great many features but in reality are very inefficient. Keeping this brief introduction about the inherent drawbacks in current human centric systems design in mind the following section provides a discussion on the scope of this dissertation.

1.1. Scope of the Dissertation

In the previous section there was a brief discussion about how human centric systems are increasingly becoming popular and how their design comes with inherent drawbacks. This dissertation focuses on tackling one such drawback inherent in today's human centric systems; the problem of interruptions caused to the user by her computing system, environment or electronic device. This problem while common (and generally ignored) today, will worsen in the near future with the growing popularity of intelligent feature laden phones, smart homes and other pervasive and situation aware systems. The reasoning behind addressing this problem is that interruptions (however disruptive) cannot be eliminated since they are necessary for a variety of reasons such as interaction, notification, updates, and data exchange. Thus, they need to be channeled in a manner that is less disruptive to users' activities. The scope of this dissertation is therefore confined to developing techniques to manage interruptions effectively. Two diverse scenarios namely the cell phone and the virtual office secretary (gate keeper) are considered due to their growing demand (and related interruption causing drawbacks) in today's society. The majority of the work focuses specifically on developing learning models for both scenarios that channel interruptions such that they are no longer disruptive. These models are incorporated in simulations to test their validity and interruption handling capabilities. Since the goal is to eliminate the disruptive effects of interruptions and make systems more human centric, a part of this work also focuses on incorporating knowledge from end-users directly into the system design process to make them more tuned to user needs. An outline of this document is presented below.

1.2. Dissertation Outline

Chapter 1 provides a brief introduction to the dissertation with a detailed outline, scope and limitations of the dissertation, and its main contributions.

Chapter 2 provides a detailed background and terminology of various areas under consideration and specifically human centric systems, context awareness, and user interruption.

Chapter 3 presents the problem being considered namely handling user interruption caused by system generated events, notifications and errors. It discusses in brief the effects of interruptions and the reasons for addressing this problem.

Chapter 4 discusses the existing solutions that have been used for handling user interruption such as system or user driven approaches and rule based or learning approaches. It then presents the proposed approach towards handling user interruption by employing learning and usability engineering techniques specifically field studies, surveys, and machine learning using decision trees.

Chapter 5 presents a variety of application domains and scenarios that are prone to system generated interruptions. The research narrows to four scenarios and provides a discussion on two of these which are chosen for further development and simulation.

Chapter 6 is the core of the dissertation in that it presents initial design and analysis activities for simulation of the two scenarios introduced in chapter 5. A detailed discussion of the design approach used, activities such as field studies, user surveys, survey analysis, and training data collection is also provided.

Chapter 7 continues this discussion by elaborating on the simulation aspects of this research. Initially, the architecture for the simulation of the two scenarios is

presented with a discussion on how this architecture is implemented. This chapter looks at various enabling technologies such as Java programming tools, machine learning software, handheld programming tools and devices, etc. that were employed in developing this simulator. Towards the end a summary of experimentation and testing activities of the two scenario simulations is presented.

Chapter 8 presents detailed evaluation and validation studies of the simulations. Activities such as training data analysis, simulator performance testing and attribute selection are presented. The latter part of the discussion concentrates on end-user validation of the scenario simulations based on comparisons of user preferences and simulator results.

Chapter 9 discusses in detail implementation issues involved in the design of interruption aware and context aware systems. A discussion on general design issues as well as enabling technologies is presented.

Chapter 10 concludes the dissertation by presenting a summary of the work done along with the main contributions of the research. A discussion on the future work and possible directions in the areas of human centeredness, context awareness and interruption handling is also presented.

Appendices and bibliography appear at the end of chapter 10, and include material such as user surveys and rulesets which were used in research activities.

1.3. Research Contributions

The research focuses on several important issues in the areas of human centeredness, context-aware systems and user interruption. Research conducted on

human centeredness and context-aware systems provides an important foundation for the problem of user interruption being tackled as a part of this dissertation. The following list represents the major contributions of this work:

- A thorough articulation of the fundamentals in the areas of human centeredness, human centric design and human-systems integration based on the study of various research efforts, relevant design issues and current limitations. Such an understanding will be helpful in generating standard design procedures such as frameworks, methodologies and the like which will assist designers in incorporating human centric functionality in their designs.
- A study of context-aware systems viewed from the perspective of human centric design and their resulting drawbacks and limitations. One such drawback, namely that of system generated user interruption is addressed in this dissertation. This is especially important given the growing popularity of pervasive devices and context/situation-aware functionality in applications like mobile devices, smart homes, and virtual assistants.
- A comprehensive study of system generated interruptions is presented which adds to the current literature on interruptions. Existing solutions are studied and illustrated with the help of a well defined taxonomy. These studies can serve as effective tools for designers planning to focus on interruption handling.
- A method to handle user interruption caused by system generated events, notifications and errors. The method proposed in this work employs information elicitation and machine learning techniques to gather information and learn about user contexts in order to handle system generated interruptions in an appropriate manner.

- Simulation of such a method by considering two popular scenarios, namely the cell phone scenario and the virtual secretary scenario. These simulations show the need for effective initial modeling, simulation and prototyping stages before any new or innovative designs are implemented.
- The use of methods such as experience sampling, field studies and surveys in order to gather accurate information about various design essentials such as training data, attribute selection, attribute alternatives and so on. The research shows that such user studies and data collection are very important towards effectively incorporating end user requirements and preferences in system design.
- A design process that clearly shows how activities in the process such as requirements analysis, surveys, field studies and data analysis are essential for developing highly effective simulations and models which can then be converted into prototypes or actual implementations. Such a process suggests a need for integration of various technological and human oriented issues during the initial design and simulation phases.
- Implementation issues involved in the design of context and interruption aware systems. These issues are specific to the scenarios in consideration but also may be generalized and expanded to include other applications. These issues cover enabling technologies, infrastructure, environments and the underlying limitations of each given the current state-of-the-art.

CHAPTER 2

BACKGROUND AND FUNDAMENTAL RESEARCH

This section discusses in detail the background and fundamental research that was performed in understanding various principles and technologies in the areas of human centric systems, context awareness and user interruption. In doing so it not only provides common terminology and past work from each field, but also presents ideas, illustrations and methodologies that help gain better understanding of these areas so that designers can incorporate better human centric functionality in their designs. Figure 1 shows how this section narrows down from the general area of human centeredness to context aware systems and then to the underlying problem of user interruption faced in context-aware systems.

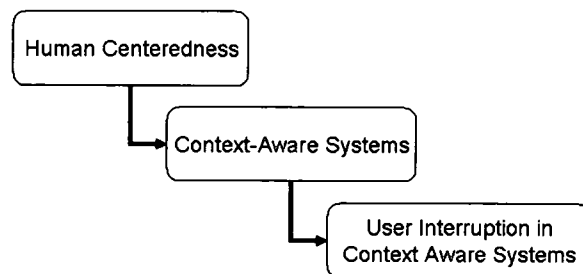


Figure 1. Background Discussion and Fundamental Work in Three Areas Relevant to this Research

2.1. Human Centeredness

The success and wide acceptance of systems designed today and in the near future will depend greatly on their ability to provide human centric work and interaction capabilities. To develop such systems we need to fully understand the concept of human centeredness. What do we mean when we talk about terms like human centeredness or human centric systems, user friendliness and human computer interaction? Many engineering and human factors disciplines use these terms interchangeably and it is quite easy to misunderstand terminology in one field from the other. Currently there is considerable research being conducted in different areas of engineering and human factors such as artificial intelligence, affective computing, mobile and wireless computing, usability engineering and so on. The main goals in these areas are to make the system or device a part of the user's environment, give it enough human-like qualities and make it responsive to the user and her surroundings. This is being done by combining a host of human factors and engineering issues. Research on affective and intelligent computing at the [MIT Media Lab] [BlueEyes] [Oz Project] has been focused on emotional intelligence: including the ability to recognize, model and understand human emotion; to appropriately communicate emotion; and to respond to it effectively. Considerable research is also being conducted on identifying and incorporating human factors issues in system design. Important factors being considered in this regard [Bradner et al, 1998] [Kellogg et al, 2001] are social interaction, group work and behavior, culture, and organization. In the areas of pervasive and mobile computing some important academic research projects such as Aura at Carnegie Mellon University (CMU) [Aura], Endeavour at the University of California at Berkeley (UC Berkeley) [Endeavour], Portolano at the

University of Washington [Portolano], etc. are being conducted using pervasive and ubiquitous devices, smart homes, wearable computing systems and so on. Research [Oviatt et al., 1997] [Flanagan et al., 1998] [Raisamo, 1998] on the use of multimodal interfaces and systems for a variety of purposes continues to expand with advances being made in various fields such as machine learning and interface design. Research on human centeredness has come a long way in making it easier for users to interact with systems by addressing usability, ergonomics, communication and interface issues. Other areas such context and situation awareness [Abowd et al., 1997] [Dey et al., 2004] are also making it possible for the systems to recognize and appropriately respond to users behavior, surrounding situation and context.

It can be noticed that there are a variety of research areas, technologies and principles being employed in the design of human centric systems today. Due to its importance in this research effort an attempt is made to distinguish between human centeredness, human computer interaction and user friendliness and to address the different perspectives involved in the design of human centric systems.

For the purposes of this research and other related projects, human centeredness was defined as follows:

“The ability of a computing system to effectively support the user in various activities, as well as to appropriately respond and adapt to the user’s needs, surroundings, and constraints.”

Consequently, our definition of a human centric system is:

“A system, device or computing environment, which provides appropriate communications, resources, services and support infrastructure to allow users to perform various activities effectively and efficiently.”

Human Computer Interaction (HCI) has been defined as follows:

“Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.” [ACM SIGCHI]

While we define user friendliness as:

“The ability of a system or device to allow different kinds of users, both novice and experienced to easily and effectively perform a variety of activities, and to provide them the necessary information and support required.”

Thus we notice from these definitions that human centeredness is an all-encompassing term. It includes human computer interaction which essentially addresses issues that deal with the interaction between users and their computing environments, whether they are standalone PCs or distributed systems.

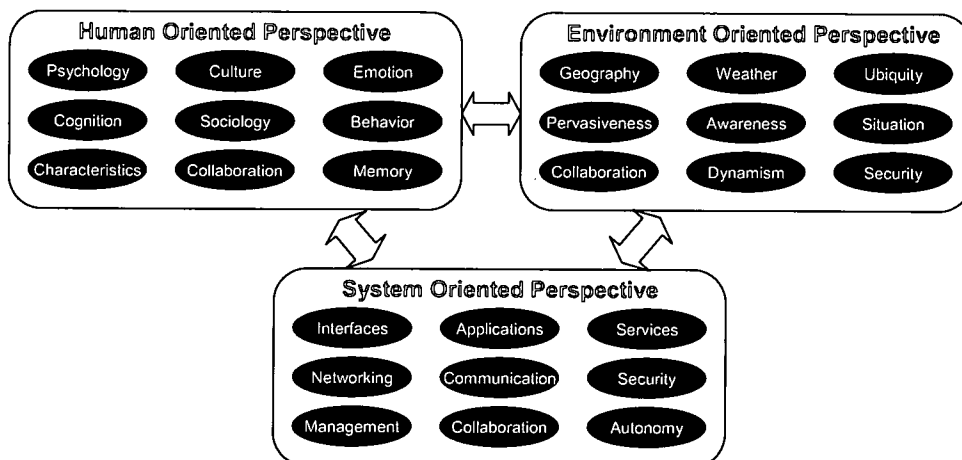


Figure 2. Three Perspectives for Human Centric System Design

It also includes human factors because it addresses users' needs, behavior and characteristics. The goal of human centric design is to address relevant system and human

issues in order to make machines easier for users to work with. Figure 2 illustrates this idea better by looking at human centeredness from three broad perspectives.

The figure illustrates that human centric system design can be pulled into any of three directions depending on design goals, biases and other criteria. Unless end user based design is effectively taken into consideration the result may not necessarily be human centric in nature. What is required is a well defined design process such as a framework-to-methodology-to-architecture approach in order to take all relevant issues into consideration. Figure 3 represents a preliminary framework integrating state-of-the-art enabling technologies with relevant human centric design criteria. Several important human centric design criteria such as culture, sociology, psychology and cognition, situation awareness, interfaces, security, etc. have been considered.

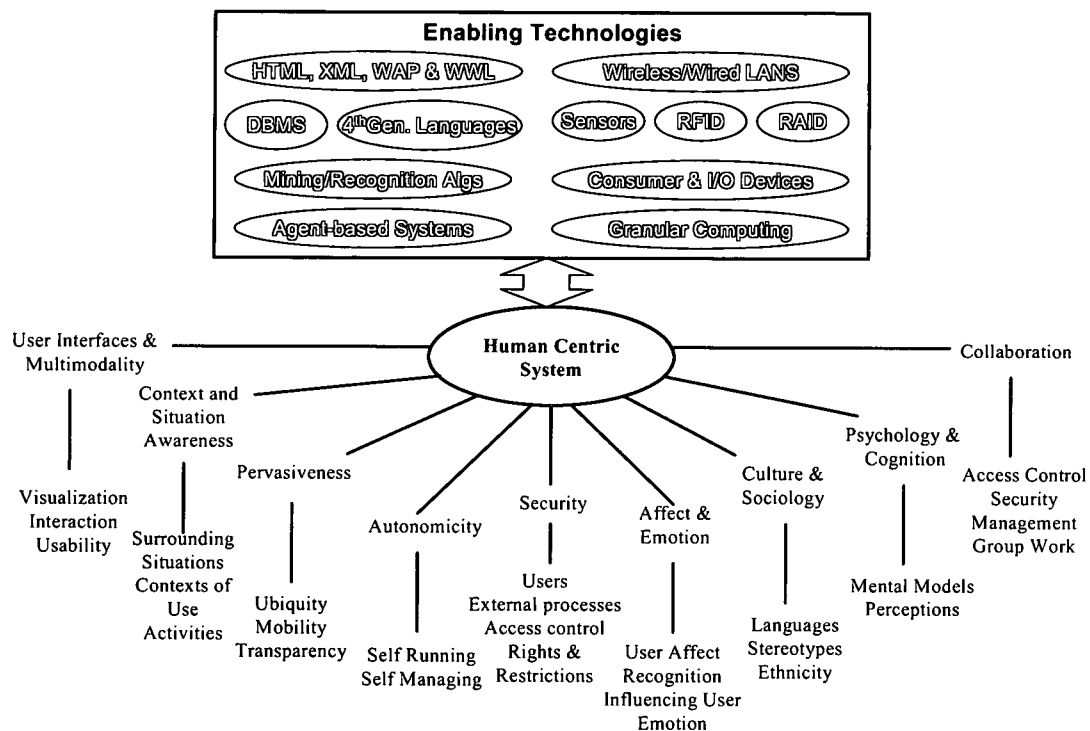


Figure 3. Preliminary Framework for Human-Centric System Design

Based on application and scenario requirements, user preferences, environmental considerations, and other factors, some or all of these criteria could be implemented using the technologies listed. Such frameworks are necessary to select design criteria and address relevant issues and requirements before proceeding with architecture design or modeling. In the later chapters we will notice how this research focuses on a specific design process in order to develop architectures and models of context and interruption aware systems. In the following sections the focus is on fundamental issues and terminology in human centric design areas that are specific to this research, namely context aware systems and user interruption handling.

2.2. Context-Aware Systems

The main principle of context-aware computing systems is to adapt services or interfaces provided to their users by maintaining awareness of their surrounding situation (also termed as context). This is typically accomplished by applying machine learning or rule-based techniques to the information collected from various situations and contexts by means of a variety of sensors. To understand context-awareness, imagine a simple system such as a laptop with hardware to sense ambient lighting. It consists of software that controls the display backlight and brightness by gathering information about the ambient lighting from a light sensor. We can describe this laptop system as being context-aware because it is aware of the local conditions and modifies or adapts an interface to suit the current situation. For the purposes of this research the following definition of context has been used:

"Any information that can be used to characterize the situation of an entity, where an entity can be a person, place, physical or computational object." [Dey et al., 1999]

In other words context can be a person, her behavior, characteristics, surrounding environment, system properties or even events. Any data that provides information about these entities can be termed as context. Therefore, a context-aware system is:

"A system, device or environment that utilizes contextual information in order to provide appropriate responses and services to the end user."

Hence a system utilizing geographic information can be termed as a location-aware system, while a system utilizing situational information can be termed as a situation-aware system and so on. In general the synonyms for context and context-awareness are [Schmidt et al., 1999] circumstance, situation, phase, position, posture, attitude, place, point, terms, regime, footing, standing, status, occasion, surroundings, environment, location, and dependence.

The concept of context-awareness has been around for a long time albeit under different names and areas. For example, Bolt's 'put that there' system [Bolt 1980] involved simple use of gesture and voice recognition to identify objects on the screen and move them. The context here was low level and was supplied via classification of voice commands and gestures into various high-level categories such as pointing to 'that' and 'there'. Other examples of such systems (albeit more advanced) include Brightboard [Stafford-Fraser et al., 1996], Quickset [McGee et al., 1998], olfactory nose-tracked displays [Yanagida et al., 2003], etc. Such systems used low level context and fell under

the category of areas known as multimodal systems. Over the last decade however with the arrival of pervasive, mobile and wireless computing, context in the form of location, network, hardware and so on is being considered in addition to pointers, gestures, and voice. Since the mid-nineties a lot of research continues to be performed on location-aware and situation-aware mobile and wireless devices such as [Want et al., 1992], [Asthana et al., 1994], [Beadle et al., 1997], [Barkhus et al., 2003], and [Xiaohang, 2003]. Context-awareness is also moving to conventional desktop systems and other non-conventional areas such as wearable and appliance computing.

Until the last decade or so there was very little research in converting low level information into knowledge that could be used for a variety of purposes. However with the emergence of context-aware systems as a popular area, considerable research [Bederson, 1995], [Abowd et al., 1997], [Biegl, 2000], [Marmasse et al., 2000] is being performed specifically in such areas as context-recognition, knowledge creation, dynamic and responsive systems. Some examples of context-aware systems developed since the mid-nineties include [Dey et al., 2004], [Xiaohang, 2003], [Chen et al., 2004], [Campbell et al., 2002], [Lockerd et al., 2003], and [Sharon et al., 2003]. Unique environments are also being researched [Meyer et al., 2003] that bring the concept of context-awareness literally into our homes. Overall, it has been seen that the notion of context-awareness is spreading into areas of computer and engineering science applications with the fundamental aim of providing awareness and making the systems more intelligent, responsive and reliable. It has also been argued by some researchers [Erickson, 2002] that context-awareness has not been as successful as it was initially expected simply due to the fact that the implicit direction so far has been to design computing systems that do

almost all activities for the user. This direction has been noticed time and again in AI research [McDermott, 1981] and Erickson suggests that the user needs to be in the usage loop in order to develop an effective context aware system. In the following section a discussion about how contextual information can be used to modify existing information or even create new and previously unknown knowledge is presented.

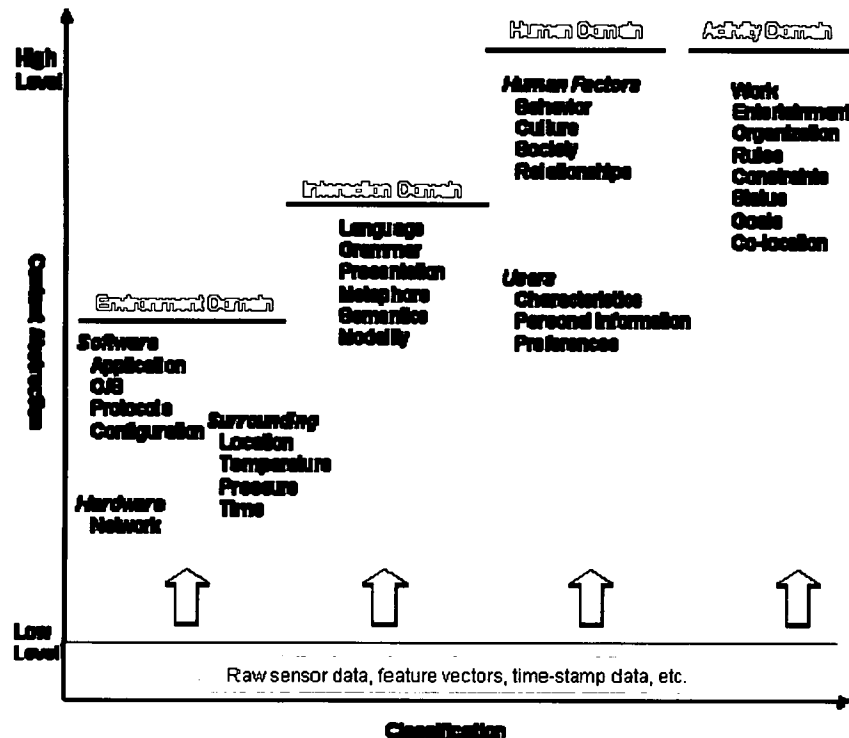


Figure 4. Context Classification Taxonomy

Figure 4 shows the context classification taxonomy with the x-axis showing the four different information or data categories and the y-axis showing the abstraction from low level data to high level knowledge. It may be noticed that raw data coming from input sensors and other hardware or software forms the lowest level of abstraction. Next come the feature vectors, cues and time-stamped data which are nothing but processed

raw data. These by themselves hold no information or knowledge and thus are also a form of raw data. This low level data and information is used by specific mining and learning techniques to generate high level or abstract contexts such as knowledge about user behavior, her current physical or emotional state, state of the surroundings, the tasks that the user or system is performing.

Figure 5 shows a generic process of context-aware adaptation that starts with the recognition of low level sensor inputs and ends with the adaptation of an interface or service. The basic idea here is to have multiple sensors provide context-sensitive information to intelligent algorithms which create high level knowledge, take decisions and perform actions. These intelligent algorithms essentially classify sensor data [Dey, 2001] into one or more categories of contexts that are at a higher level of abstraction. For this they require prior training or in some cases may be completely autonomous.

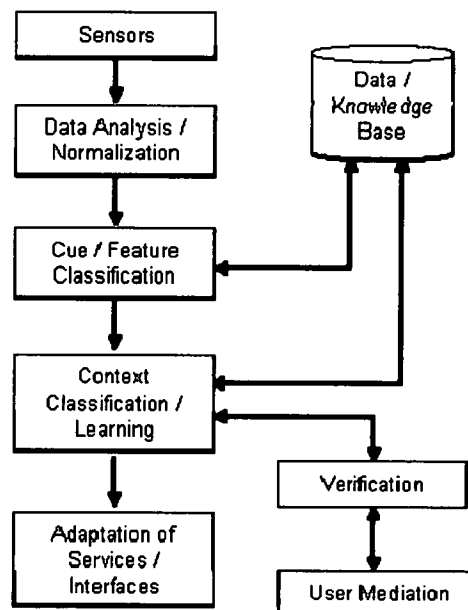


Figure 5. A Generic Process of Context-Aware Adaptation

This process continues until knowledge or context that is sufficiently understandable by humans is produced [Dey, 2001]. Algorithms used in this process employ rule-based, example (case) based, and inference (learning) based techniques to generate such knowledge or context [Barkhus, 2003] [Schmidt, 1999], [Van Laerhoven, 1999]. Appropriate actions are taken when a specific context or situation is determined, or alternately appropriate triggers may be sent to applications for performing actions.

Consider an example where cameras and microphones provide video and audio data to the system. If the system has been designed to recognize eye or lip movements and specific words or expressions it would categorize closed eyes and yawns as the 'user being bored or sleepy'. Depending on what the current situation is, whether 'exams week' or 'no specific urgency', the system may prompt the user to keep working to finish the task on time or to take a break. For all this to work effectively the accuracy of the recognition process must be extremely high, i.e. the system should be able to distinguish between subtle changes in eye and lip movements, or similar sounding words, etc. This accuracy may be improved by the type of algorithm or technique in use, the available knowledge in the form of classification parameters, or even the hardware and software used for input. This 'frame problem' [Hayes et al., 1971] also comes into the picture when the recognized input needs to be categorized into one or more of a wide range of possible contexts. So the more generic is the scenario is the larger is the number of possible contexts as we will see later during the design part of this research. Of course, not all possible contexts are valid or even realistic. Hence the complexity can be greatly reduced by modeling the possible use cases with their system responses.

2.3. User Interruption

In the previous sections a detailed discussion about human centric systems and context awareness was provided. It is evident from this discussion that providing human centric properties is a complex task that involves the integration of techniques and tools from various disciplines. While human centric functionality provides many benefits to users it may also have some inherent drawbacks. For example, a software suite designed for users in one country may not work satisfactorily in a different country due to factors such as language, jargon, culture, social and religious constraints, etc. Similarly an interface designed for a majority of intended users may not be suited for the minority owing to physical, cognitive or other limitations. One such drawback faced by devices and computing systems is that of user interruption. There have been a number of studies on the general effects of interruption both human and system generated [McFarlane, 1999] [Hudson et al., 2003] [Franke et al., 2002] [Rouncefield et al., 1994]. Research in user interruption handling, resolution and awareness on the other hand is quite recent and not extensive [Hudson et al., 2003] [Siewiorek et al., 2003] [McCrickard et al., 2001], and efforts concentrate on rigid implementations and static rules that are not compatible across domains. More on the related work on interruptions and interruption handling will be discussed in the section on solutions in chapter 4.

For now we remain focused on describing the fundamentals of user interruption, in particular one aspect of user interruption that is caused due to a variety of system generated events. A clear distinction is made between interruption to users by humans and interruption to users caused by systems. Here the main causes of user interruptions in systems are studied, along with their characteristics and categories.

For the purposes of this research, the following definition for interruption is used.

"...Interruption is the process of coordinating abrupt change in people's activities." [McFarlane, 1997]

"Postulate 1: This abrupt change involves one or more of a person's modes of activity: (1) cognition (2) perception, or (3) physical action." [McFarlane, 1997]

This definition represents a general unified definition for interruption that includes interruption caused to humans by other humans, and interruption caused to humans by systems or devices. Since the focus of this research is on system generated interruption, the following working definition for interruption is provided:

A break in the continuity or uniformity of a user's activity, focus, or cognition, caused by system generated events during human-system interaction.

Having defined system generated user interruption the following paragraphs look at the main causes of such interruptions, their important characteristics, and the different types of interruptions commonly in occurrence today. This discussion is illustrated by Figure 6 which describes the fundamentals of system generated user interruption, namely the causes, characteristics (or properties) and categories.

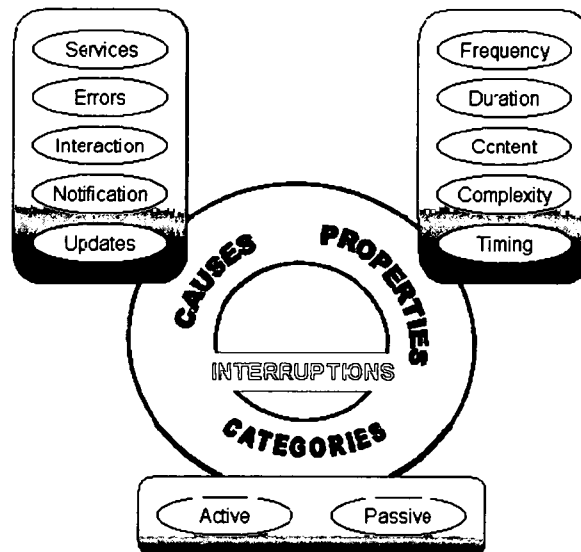


Figure 6. The Basic Fundamentals of System Generated User Interruption

System generated interruptions can be caused by a variety of factors [Czerwinski et al., 2000] [Rouncefield et al., 1994] [Hudson et al., 2003] [Gillie et al., 1989]. Given the diversity of electronic devices, computing systems and environments being used today in equally wide ranging scenarios, the following list of factors summarizes the causes of interruptions.

- System adapts a service or an interface in response to certain events. This is quite common in GUI based systems or devices where events or actions can trigger the change in screen based GUIs or services that the system is currently providing. For example, the Microsoft Windows screensaver mode where the system goes to 'sleep' based on a predefined setting. This could cause interruption if the user was listening to music or reading a document on screen. In the same manner, cell phones usually switch to a different screen whenever there is an incoming call. This could be mildly intrusive if the user was in the middle of some screen based activity.

- System actively responds to an error, malfunction, or anomaly in the functioning of tasks. Interruptions caused by errors are usually non-avoidable depending on the severity of the error. Critical system errors for example leave the user with no choice to save data or revert back to the pre-error situation. Mild errors are usually indicated with option GUIs that provide users with the option to either immediately handle the error or to handle the error at a later time.
- System notifies user about updates, past communication, messages, errors or other events. This is usually random and can cause the most interruption because it does not leave the user any time to prepare. Unlike interruptions caused due to errors however, users have the ability to respond to or neglect the interruption and in the process save their current activity for later continuation.
- System indicates of possible occurrence of interruptions so that the user has adequate time to prepare for the interruption. This is generally not as common as the other types of causes because it occurs only when timing and periodic notification are essential. In either of the two cases users typically take note of such interruptions and disregard them to continue with their current activity.
- System interactions with the user while performing tasks and responding to user queries. These are not considered interruptions unless a user is performing multiple tasks at a time and certain events in a background task force it to interrupt the user when she is performing another task. In this case, a user's cognitive load increases because she has to process information about multiple tasks and this can lead to errors on the users part.

Based on studies of interruptions in computer based activity and decision making [Speier et al., 2003] [Czerwinski et al., 2000][McFarlane, 1999] interruptions can be characterized by a variety of factors. The work by Speier [Speier et al., 2003] best summarizes the factors that characterize interruptions as follows:

- Frequency: this indicates how often the interruptions occur and depending on the users activity can be quite distracting.
- Duration: this indicates how long a particular interruption lasts and again, depending on the type of activity the user is performing, can be distractive because they increase the user's cognitive load.
- Content: this indicates what the interruption is about. In other words, is it a query that requires information? Is it a simple notification? Or in the worst case, is it a critical error that requires the user to reboot a device?
- Complexity: this indicates how much cognitive effort the user has to put in handling the interruption. Usually queries and active interactions are more complex to handle than notifications or messages. In some cases the user may be required to perform additional tasks to handle the interruption and return to the original task at a later time.
- Timing: this is probably the most critical factor because usually it cannot be controlled, unless the user has the ability to set certain parameters like time or duration.

Based on the type of system generated interruptions, this research categorizes them into two types, namely Active Interruptions, and Passive Interruptions. Although there are many ways possible for categorizing such interruptions (type, content, situation,

etc.), it was felt that a categorization that is based on the user's perceptibility of an interruption would best suit our requirements. Below we look at each category in brief.

- Active interruptions engage the users' senses in a manner that causes the user to break the continuity of her task and address the interruption. Such interruptions are highly perceptible in nature, and the user may recognize them via sound, images, touch, skin response, or even pressure. The user eventually may not even handle the interruption, but the fact that her activity was cut short means that she was actively interrupted. Notifications, messages, queries, and dynamic changes that require immediate or mandatory user response fall under this category. Other kinds of interruptions with modalities such as voice, beeps, animations, temperature, pressure, etc. are also examples of interruptions that are active and highly perceptible by humans.
- Passive interruptions on the other hand may not engage the users' senses in a manner that causes her to break the continuity of her task. These type of interruptions are peripheral in nature, and to a certain extent do grab the users attention, but may not do so sufficiently enough to make her break the continuity of her task. Such interruptions have very low perceptibility. Examples of passive notifications are certain types of notifications, indication messages in the form of light or LED's, vibration on cell phones, heat, and so on. Passive interruptions provide users with a certain degree of flexibility, either in terms of handling the interruption, or sensing it in the first place.

In this section fundamental research and terminology in the areas of human centeredness, context-aware systems, and user interruption was discussed. In the next chapter we look at the problem statement and reasons for selecting this problem.

CHAPTER 3

PROBLEM DEFINITION: SYSTEM GENERATED USER INTERRUPTION

In the previous sections there was a detailed discussion on the background and fundamental research in the areas of human centeredness and context-aware systems. It was noticed that the increase in human centric functionality in system design can have many drawbacks if not conceived properly. One such drawback that was discussed is system generated user interruption. We briefly considered the causes of such interruptions, their properties and main categories. Here we will delineate this issue as an open problem that needs to be addressed by looking at important reasons and effects.

3.1. Problem Definition

Below, a concise definition of the problem tackled as a part of this research project is presented along with the reasons for selecting this particular problem.

One of the main drawbacks of human centric and context-aware systems today is that they cause frequent and untimely interruptions to users. Interruptions cannot be eliminated because they are an essential part of the system's dynamic and responsive nature. In spite of their importance they need to be channeled because they can cause hindrances and disruptions that range from mildly annoying to catastrophic. Currently there is a lack of extensive research in interruption handling, and there are few solutions that specifically focus on solving this problem.

This work attempts to address the problem by proposing a new solution that handles interruptions by using user driven information elicitation in combination with machine learning algorithms in order to develop accurate 'interruption-aware systems'. Further discussion of these solutions is presented in the next chapter. In the following paragraphs the main reasons for selecting this problem are explained.

3.2. Reasons for Selecting the Problem

One of the most important reasons for addressing the problem of system generated user interruption is that it commonly occurs in many of the electronic devices and computing systems that are in use today [McFarlane, 1997]. Good examples of such systems are control stations, airplane cockpits, collaborative systems, instant messaging tools and cell phones. Furthermore, with the growing popularity of pervasive and context-aware systems such as smart homes, virtual assistants, intelligent systems, etc. this problem is likely to become more significant, strengthening the imperative for a comprehensive solution. To summarize, systems being designed today are highly:

- Dynamic: they are able to adapt to situations, events, conditions and preferences very quickly and very easily thus distracting the user more frequently.
- Responsive: a variety of sensors, media and other tools make them respond to user interaction in many ways forcing the user to be more attentive and alert, neither of which is consistently possible for human beings.
- Intelligent: current technological advances in machine learning and awareness allow systems to become independent and autonomic to a certain degree, thus reducing users control over them.

- Multimodal: the use of multiple communication modalities for human computer interaction such as speech, video, gestures, and gaze overloads the users senses.
- Pervasive: embedding systems in a user's day-to-day surroundings makes them more disruptive and invasive vis-à-vis her activities or interactions with other people in a social setting.

These features make system users prone to frequent, untimely, sudden or even erroneous interruptions. The problem with system generated interruptions is that they cannot be completely eliminated since they are essential for purposes such as interaction, querying, information exchange, notification and updates. At the same time they cause hindrances that range from mildly annoying to catastrophic [Bainbridge, 1984] [Van Solingen et al., 1998] [Madhavan et al., 1993]. It is therefore necessary to evaluate them and handle them in a manner that is suited to the end-user.

There have been a considerable number of experimental and field studies performed in attempting to measure the benefits and drawbacks of interruptions. Some of the early works [Gillie et al., 1989] [Speier et al., 2003] measured cost of interruptions in terms of recovery time, task execution [Adamczyk et al., 2004], accuracy of post-interruption tasks [Latorella, 1999] and even evaluation of learned models [Horvitz et al., 2003]. There have been other efforts that focus on the effects of interruptions on users in terms of physical and mental stress [Liu, 2004]. While such research continues there is an overall agreement that the hindrances caused due to interruptions largely outnumber their benefits. This has also been noticed by studies [Cohen, 1980] [McFarlane, 2002] [Bailey et al., 2001] [Gillie et al., 1989] that are based on experimental or field quantitative analysis of data obtained during interruption of users activities.

These research efforts have concluded that interruption if not managed properly may result in one or more of the following hindrances:

- Delays: loss of time, work schedules and activities based on the frequency, complexity and longevity of the interruption. This is an important factor if time is an important issue in users' activities.
- Errors: manual or random errors caused due to the distraction of users by the interrupting entity. Errors can be mildly annoying or catastrophic in nature. Errors can cause additional hindrances such as delays, and loss of continuity.
- Increased Cognitive Load: this occurs when the user has to switch from his current activity in order to handle the interruption. Depending on the complexity of the interruption the user may face an increased workload in addition to his original activity.
- Loss of Continuity: is a factor when a user is performing multiple tasks, or when the interruption requires the user to be away from her original task for a considerable period of time. This creates delays and causes an increased cognitive workload on the user because she has to get accustomed to her original task again.
- Loss of Focus and Concentration: this is a factor when the interruptions are untimely or sudden, and use active multimodalities for interrupting the user. Common examples are phone ringers, and alarms.
- Socio-psychological Effects: such as annoyance, anger or frustration can also be caused by interruption. This may lead to the user making subconscious errors in her activity.

- Cost: this factor is usually a secondary effect of interruption as it is evaluated by the amount of damage caused in terms of loss of time, work schedules and equipment cost, and may be measured by quantifying some or all of the above effects. Cost may sometimes not become noticeable until long after the interruption has occurred.

To summarize, interruptions caused by systems and devices often cause hindrance to the end-user and this problem continues to grow with the increasing popularity of human centric devices and systems. Interruptions cannot be eliminated and need to be handled appropriately to suit the needs and considerations of the end-user. Therefore, this research effort focuses on the development of techniques that manage interruptions by incorporating requirements and preferences of users with machine learning tools in order to develop effective 'interruption-aware systems'. There is some research being conducted in addressing this problem and a few solutions have been implemented. These will be discussed in the next chapter along with the proposed approach.

CHAPTER 4

EXISTING APPROACHES AND PROPOSED SOLUTION

In the previous chapters the discussion primarily focused on the background, fundamentals and drawbacks in the areas relevant to this research effort. It was here that the problem of system generated user interruption in today's systems and devices was identified. Reasons for selecting this problem along with its causes and characteristics were also explained. This section concentrates on providing ideas and solutions for tackling the problem of user interruption. Initial discussion is on related work and existing solutions in this regard after which the proposed approach is presented in detail.

4.1. Existing Approaches

Figure 7 illustrates a taxonomy that represents current solutions for handling system generated user interruption. As the figure shows, current efforts can be broadly categorized into two types, firstly solutions that focus on users themselves for handling such interruptions, and secondly solutions that focus on enhancing or modifying aspects of system design in order to reduce or eliminate the disruptive effects of such interruptions. In the following paragraphs each of these techniques is discussed based on

examples from current and past research efforts. Here, the advantages and limitations of these techniques are also presented.

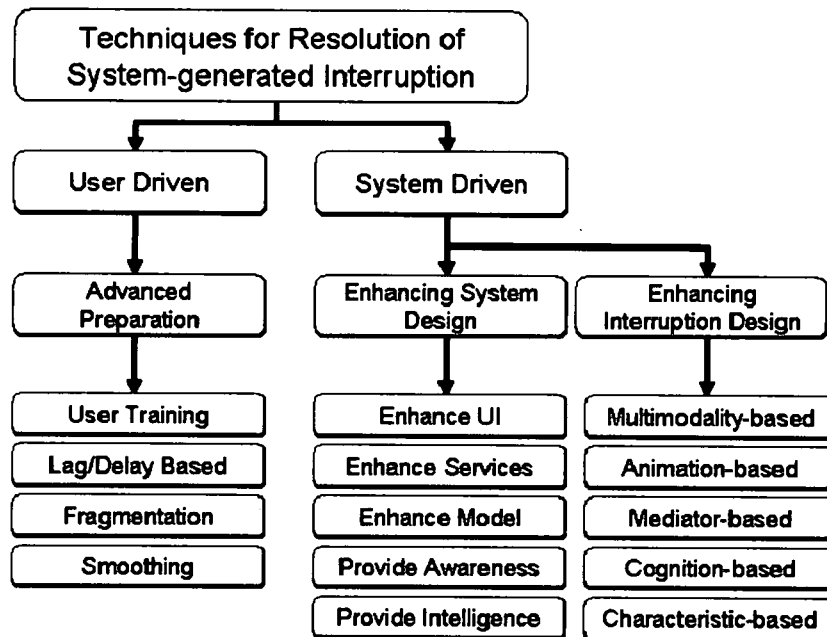


Figure 7. Taxonomy of Solutions for Handling System-generated Interruption

4.1.1. User Driven Approaches

User driven approaches for interruption handling mainly focus on providing users with additional support, guidelines and information in order overcome interruptions without adversely affecting their task performance. Based on the relatively limited research in domains such as air traffic control, business environments and so on, user driven approaches to interruption handling can be broadly categorized into four types; namely training-based, lag or delay based, fragmentation-based and smoothing. Each of these categories is based on how the user responds to the interruption by means of

protocol based training, task scheduling, advance preparation during pre-interruption intervals and transitioning between original tasks and interruption tasks.

In the first case research [Gillie et al., 1989] [Franke et al., 2002] [McFarlane, 2002] suggests that advanced training and preparation can be useful for handling interruption in human computer interaction by employing rehearsal, memorization or context recovery for negating some effects caused by memory loss of the original task. Studies and experiments in *prospective goal encoding* and *retrospective rehearsal* [Trafton et al., 2003] [Brandimonte et al., 1996] [Goschke et al., 1993] [Patalano et al., 1997] indicate that users can employ interruption lag to prepare for their original tasks suggesting that training and lag based methods work in combination to improve the performance after interruptions.

Fragmentation of activities around interruption activities or vice versa also can provide users with some flexibility and reduce hindrances. Some studies and evaluations [Bailey et al., 2001] [Mark et al., 2005] [Monk et al., 2002] [Bannon et al., 1983] have been conducted which propose the division of tasks based on their phases of planning and execution such that interruptions can be handled more easily. Such studies also show that users' tasks are typically interleaved and it is possible to identify specific points in multiple activities in order to interrupt the user for a least possible negative effect.

On the other hand, research [Detweiler et al., 1994] [Altmann et al., 2004] [Hansson et al., 2001] on smoothing by offering early warnings or cues indicates that considerable benefit can be achieved in situations where the interruptions themselves are complex and time consuming, or socially disruptive as in cases of mobile devices. Cues

and warnings either subtle or active are important because they provide users with advance preparation time to more effectively handle the interruptions.

These studies show that while user-driven methods can be employed as a first response towards handling interruptions, they do not necessarily produce a significant increase in user performance or productivity in post-interruption activities. The following are the major drawbacks that have been summarized by studying the works on user-driven methods above.

- The term user-driven interruption handling is a misnomer in that, interruptions themselves are not resolved. It is the user that has to take active steps in order to avoid hindrances caused by interruptions. These studies have shown that while hindrances can be avoided, more time and cognitive effort is spent in returning to the original tasks and results in performance improvement are marginal at best.
- User driven approaches involve a considerable increase in workload in users. This is due to the fact that the users have to perform specific pre-interruption and post-interruption activities in order to make the interruptions smooth. These activities may involve memorizing or trying to remember the state of the original activity, in physically making notes or even preparing for impending interruptions.
- Approaches such as fragmentation or task switching depend on the frequency of interruptions in order to be effective. Studies have shown that if the frequency of interruptions is low, switching tasks may prove to be beneficial because the users are performing different tasks periodically and hence are able to remain refreshed.
- These approaches incur additional costs in terms of time, user training, capital and so on. This has been noticed especially in training employees who work in assembly

lines, warehouses, etc. The results are not always productive and managers have to apply scheduling, logistics or supply chain principles to avoid interruptions.

- User-driven approaches need to be effectively conducted. In case of multiple activities and frequent interruptions, users need to keep records of original tasks, state of each task, variables involved before handling the interruptions so that they can smoothly switch back to those tasks after the interrupting tasks have been handled.
- Some approaches such as providing cues, warnings or notifications can be helpful provided the modalities themselves do not cause anxiety, stress, annoyance or sudden switch of focus from the original tasks. If the notifications are predictive or scheduled then users can prepare for them by anticipating and working towards switching the tasks beforehand.

Overall it was noticed that user-driven approaches only partially solve the problem of interruption handling and involve considerable time and effort on the part of the user, neither of which are always possible. In the following section another approach at interruption handling is studied, namely the system driven approach, with some of its benefits and drawbacks. After that the discussion focuses on the proposed approach in this work and how it can help in effectively resolving system generated user interruptions.

4.1.2. System Driven Approaches

In contrast to user-driven approaches, there is a considerable amount of research in system driven approaches. This research however is distributed in a variety of disciplines such as workplace management, computer supported cooperative work (CSCW), user interface design, context-aware systems and artificial intelligence. Most of

this research is also relatively limited in scope because it is application specific and focuses on notifications and human computer interaction, rather than actual interruption handling. System driven approaches can be broadly categorized into two types; one where the system functionality or underlying model is modified to resolve problems due to interruptions; and second where the interruptions themselves are enhanced so that they do not cause hindrance or disruption of users activities. Both of these sub-categories are discussed with related research below along with summary observations about their limitations and disadvantages.

4.1.2.1. Enhancing System Design

The most popular and widely employed method of reducing hindrances caused due to interruptions has been to enhance the design of the system, its underlying model and its components such as user interfaces, system services and so on.

Research on enhancing system services such as communication in human computer interaction [Czerwinski et al., 2000] [Avrahami et al., 2004] focuses on developing techniques for handling interruptions generated by Instant Messaging communication. Evaluation of performance and responsiveness due to notifications conducted in such studies underscore the harmful effects of interruptions. Efforts by Abowd and Ramchurn for example, focused on providing services based on user location to handle interruption [Abowd et al., 1997] [Ramchurn et al., 2004]. Interruption handling here focuses on adapting services and interfaces based on user situation and location. Other areas of growing interest to interruption designers are those of smart homes and tangible interfaces. Due to responsiveness and dynamism in these types of

systems there is a necessity to effectively manage the services they provide. A survey of research on smart homes [Meyer et al., 2003] and work by Ishii [Ishii et al., 1997] suggests that the use of learning techniques, sensors, ambient systems and wireless technologies has enabled such systems to become human centric. As a result they have also become highly disruptive partly due to their properties and partly due to improper user training and experience in using such systems.

Interface design has been a widely targeted area for human computer interaction designers. Shneiderman [Shneiderman, 1997] discusses user interface design as the first step towards achieving effective human computer interaction. It is also a way of resolving interruptions [Bailey et al., 2000] [Harrison et al., 1995] [Lee, 2005]. These works focus on developing innovative user interfaces that are dynamic and pervasive in nature and provide awareness of situations and events so that the user is prepared for eventual interruptions. Some of these are web-based, some are pervasive and located throughout the user's environment, and some are transparent and multi-layered to provide users with the best possible means for maintaining task awareness.

In addition to user interface design and information awareness, enhancing the system model can also help in resolving interruptions and make interaction more human centric. Studies such as [Norman et al., 1986] [Horvitz et al., 2003] [McCrickard et al., 2003] discuss the need for effective recognition of users cognitive needs and limitations, attentional focus, and decision making abilities during the design process in order to make user system interaction smooth and non-disruptive. Modeling languages and interface specification tools can be employed for developing system models in order to detect possible interruption scenarios.

Sensors, learning models, decision aids, information awareness tools, etc. can also help support systems in making interaction more human centric. These can be used in integration with a variety of awareness, mining and learning techniques to generate effective interruption aware system models. Such models allow systems to be aware of events, user behavior and surrounding situations, thus allowing them to channel interruptions in a more appropriate manner. There is considerable research on providing awareness [Horvitz et al., 2003] [Hudson et al., 2003] [Yan et al., 2000] [Siewiorek et al., 2003] and intelligence [Horvitz et al., 1999 and 2003] [Gibbs, 2005] [Giveska et al., 2005]. Most of the focus here is on making systems dynamic, responsive and aware of users and their surroundings. This is an active approach wherein the system channels the interaction to suit the users' needs. Applications of these approaches include virtual office assistants, augmented instant messaging, virtual tour guides and cell phones.

4.1.2.2. Enhancing Interruption Design

Interruptions can also be resolved by enhancing the design of the interrupting entities themselves. This includes enhancing the interrupting modality, making the interruption mediator based or animation based, making the interruptions cognitively simple and easy to resolve, and interrupting based the type of interruption.

Interrupting users based on modality was studied by [Arroyo et al., 2002] and [Brewster et al., 1994] where responses by the system to users during user-system interaction were enhanced by the use of various modalities such as heat, smell, sound, vibration and light. Results in these studies and other specific research [Welch et al., 1986] indicate that active interruptions such as sound are more disruptive than passive

interruptions such as heat, light or vibration. In other words, differences in multimodal responses interrupt users to varying degrees. Other work employ similar multimodal techniques [McGee et al., 1998] for user system interaction to provide user with appropriate interaction capabilities and reduce hindrance caused due to interruptions.

Animation or graphics based techniques [Maes et al., 1994] [McCrickard et al., 2001] are also being studied for passively interrupting users. While animation has not been solely used as the interruption modality, it is often used in combination with other GUI based methods as seen in everyday instant messaging, calendaring and scheduling software.

There is very little research in the areas of mediator based, cognition based or characteristic based interruption handling. [Dekel et al., 2004] [Avrahami et al., 2004] for example employ software mediators such as agents that communicate between users and the system responses to indicate events and activities and smooth interruptions. Cognition based interruption handling involves the design of interruptions such that they can be easily understood and resolved by the user. This is usually done at the user interface level and Norman and Shneiderman [Norman et al., 1986] [Shneiderman 1997] provide substantial literature to design such interactions. Research on characteristic based interruption handling is even rarer. [Savioja, 2004] for example addresses different kinds of alarms for different types of interruptions in control room environments. In general interrupting entities are handled manually (such as setting schedules, reminders, email notifications, cell phone tunes, and so on), while automated methods are based on high level user interface or system design requirements. Examples of these are non-negotiable critical error messages, and option driven dialog boxes.

Research on system driven techniques suggests that they are more effective compared to user based techniques. However, it was also noticed that most system driven techniques are application specific and cannot be applied across domains. Many techniques also handle interruption as a part of a higher set of requirement such as awareness or intelligence and hence do not fully solve the issues related to interruptions. It was also noticed that there is negligible research on the development of frameworks and methodologies for handling interruptions and there exists no comprehensive taxonomy to summarize the methods employed for interruption handling.

Therefore as a part of this research, a taxonomy summarizing the methods employed for interruption handling is presented. The solution proposed here for interruption handling is also new and incorporates various user-based, learning and preference elicitation techniques to resolve interruption. The next section discusses the proposed solution in detail before moving on to details of the design and simulation in the subsequent chapters.

4.2. Proposed Solution

In the previous section the discussion centered on existing solutions for handling interruptions generated by the system. These solutions were categorized into user driven and system driven approaches and it was noticed that system driven solutions are generally more consistent at handling interruptions than user driven solutions. In this section the proposed solution is presented along with discussion about the approaches employed.

The proposed solution is a system driven approach, i.e., the system is designed in such a manner that it is able to learn about the situations a user is in, and interrupt appropriately such as to cause the least amount of hindrance. In order to do this, principles from human factors engineering, machine learning and software programming are employed. This approach focuses on not only enhancing system design but also the interrupting components themselves.

The problem with most approaches that employ learning, recognition or awareness techniques for interruption handling is that they focus on human activity or behavior inference which is an extremely complex task. It involves integrating a variety of components such as sensors, learning algorithms, decision tools, user interfaces and innovative computing systems. This task is further complicated by the inherent uncertainty in human activity or behavior processes and their random dependence on surrounding conditions. The focus of this dissertation however is not on human activity, behavior, or inference, although examples of such research may be found in [Simon, 1990] [Singh et al., 2003] and [Philipose et al., 2004].

This research instead aims to circumvent the aforementioned complexity of activity inference by providing a simpler solution that incorporates first hand information from users obtained via surveys, field studies and experience sampling in order to accurately recognize and learn about the interruptability of a user. In doing so, explicit activity or behavior inference is avoided while maintaining a high level of accuracy and scenario independence for interruption handling.

This solution is achieved by dividing the problem into three parts: namely a pre-design information elicitation part, an interruption-aware model learning, and a post-design preference elicitation part. Figure 8 illustrates these divisions clearly and how they are integrated in this approach. The first part gathers information about user

interruptability, day-to-day activity, behavior, etc. by means of surveys, field studies and experience sampling. This is important because it provides a foundation for generating the learning model and also helps in selecting the attributes, sensors and other tools that will be required for implementing such a system. The second part is where all the information gathered is converted into a trained learning model by employing a variety of machine learning algorithms. This model is then able to sense new events and situations, and respond in a manner that is least interruptible to the user. Since user behavior and preferences are highly variable, there needs to be a mechanism that makes the system effective for any type of user. Hence the third part of this work is required which essentially gathers individual user's preferences and refines the learned model appropriately. Machine Learning approaches provide greater advantages than static rule or case based approaches. This is especially important because the information about users, their preferences and contexts is being fed to the system at various points in the design process.

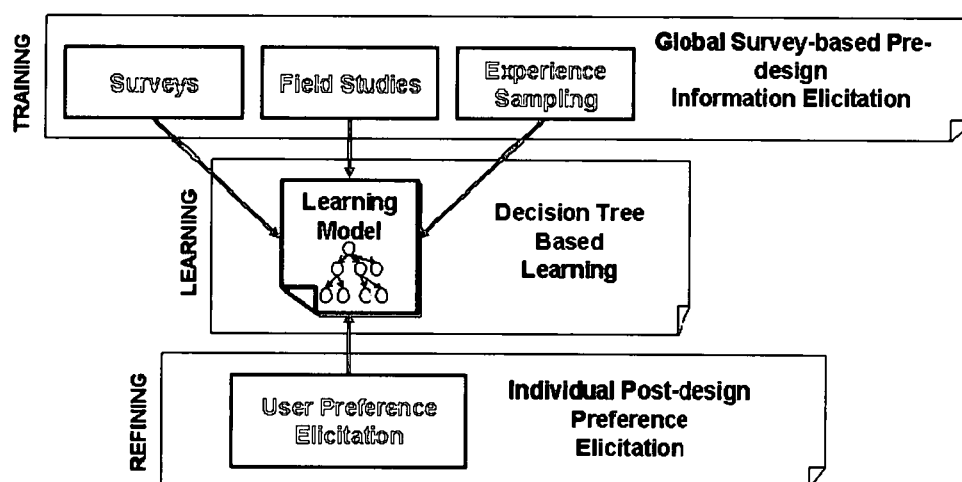


Figure 8: Proposed Approach Employing Pre and Post Design Elicitation for Learning Accurate Interruption-aware Models

Figure 8 above clearly shows how part one and part three feed into part two. This is required in order to keep the learning model accurate and at the same time represent global users' characteristics as well as individual user preferences. The advantage of incorporating user data is that sensors are not required to create training data for the learning model. User data also narrows down the implementation alternatives in terms of choice of sensors and other infrastructure requirements. In the section on design we see how our approach can be included in a typical design process. In the following paragraphs a detailed discussion on techniques used in realizing our approach is presented.

4.2.1. Pre-design Information Elicitation

Every engineering and software design process starts with an information compilation phase. Prior knowledge about application scenarios, users, environments and constraints is essential for developing an end-product that is an accurate representation of the design goals and requirements. In the case of software and computing system design such information can be actively employed during various design phases such as for personalization [Perugini et al., 2002], tailoring [Steimerling et al., 1997], and participation [Muller et al., 1993]. Each of these techniques incorporates user input in some form or another. The approach employed in this work is similar in that information is gathered from the users via methods such as surveys, field studies and experience sampling. The difference is that unlike other approaches where this information is used as a set of guidelines for designers, or in some instances for self-configuration, this approach uses the information as actual data to develop the system. In other words, the

information is used to create training data for the learning models of the interruption-aware system. For example, if surveys and field studies indicate that users prefer to switch their cell phones off during meetings, then the learning model is trained to follow that preference by choosing to switch off the phone automatically, or employ a passive interruption mode.

In this manner pre-design information elicitation satisfies a global set of users by being biased towards the majority. This is essential because it allows the eventual system that is designed to be commercially viable, and satisfactory to most users. In order to address the minority users whose requirements also need to be accounted for, the post-design preference elicitation is incorporated. This ensures that any users that do not like the current interruption handling scheme also have the flexibility to modify it. This will be discussed in the following sections.

Pre-design information elicitation using surveys, field studies and experience sampling offers many advantages for our research purposes. They are as follows:

- Allows designers to collect data about the characteristics and preferences of the design, its users, and surroundings. If activities like surveys, field studies or experience sampling are not conducted, some important issues may be overlooked, and the resulting design will not reflect proposed deliverables.
- Assists in narrowing down the design requirements and goals to a more realistic, feasible and relevant level. In some cases, it may be theoretically possible to design a given system according to listed requirements, but initial surveys or field studies may unearth possible conflicts that mandate changes to design goals or requirements.

- Allows the development of complete and accurate simulations, models or even prototypes without the use of actual components such as sensors and other infrastructure. This facilitates in an accurate selection of infrastructure and enabling technologies thus reducing the cost and effort required to make changes midway through the design process.
- Provides a statistical validity to the eventual design since most design components and functions are based on input from a user population. In case statistical majority cannot be achieved (i.e. where user responses are equally divided) post-design preference elicitation methods can assist individual users in modifying the system design to suit their preferences.
- Provides realistic and timely collection of data that is not intrusive to users' activities. Using these methods, information collected can be highly extensive and can also include rare events, conditions and cases, all of which are not always possible in experimental situations
- In many scenarios it may not be possible to deploy usability experts, sensors or other information collection mechanisms. Thus, these methods are useful when scenarios do not permit comprehensive field studies, experimental setups or sensor based information collection devices without active disruption of users' activities.

In spite of the many advantages pre-design information elicitation provides, it is still a time consuming and complex activity which requires considerable effort and labor in terms of creating surveys, sampling and field study setups, distributing surveys to user populations and analyzing returns. An alternative to this is to automate the process in one of two ways. One is to get users to enter the required information themselves, and the

other is to use devices that reside the users' environment to gather the required information. Once all required information is obtained, designers can manually analyze and incorporate results into system design or use a host of knowledge mining techniques to automate the process. We discuss these alternatives towards the end of this dissertation under the section on alternative methods for interruption awareness

4.2.2. Decision Tree-based Learning

In this section the core learning method employed for developing interruption aware systems is discussed. As seen in the section on existing solutions, learning based approaches have been employed for various purposes such as situation awareness, emotional and affective responsiveness and intelligence in order to make systems responsive and among other things sensitive to disruption of user's activity. In general learning approaches provide the following advantages compared to static rule or case based approaches:

- Available information in the form of numeric or discrete data is combined effectively to create new and meaningful knowledge that can be employed for various purposes. Existing knowledge can also be refined by adding new concepts to it.
- Have the ability recognize relationships in data that may not be distinguishable by manual or static means. This is especially useful when the information available is considerable and it is not possible for humans to account for all relationships.
- The ability to create and learn new concepts from existing data gives the system greater flexibility in responding to situations or events. This is especially important given the variability and uncertainty in users' situations, behavior and activities.

- Resulting systems become dynamic, responsive and flexible since they do not have to adhere to a pre-determined set of rules and policies. In this manner they can be easily suited to changing scenarios and domains.
- Useful in scenarios that are complex and involve large numbers of variables, rules and policies. In such scenarios learning approaches require only a subset of the entire spectrum of information to work effectively thus reducing the need for rule and policy generation, information gathering, and so on.
- Enable systems to become human centric by being aware of users surroundings, events, behavior, emotion, and so on. By having more knowledge at their disposal systems can make users' activities easier by reducing cognitive load, complexity and physical effort.

The approach employed in this research uses decision tree based learning in order to recognize users' situations and interrupt accordingly. Decision tree learning is a supervised learning method in which the fundamental idea is to create a learning model from a set of known examples (by supervision or training) and employ the model to recognize new and previously unseen examples. These examples are nothing but situations, cases or events that have occurred in the past and have a known response associated with them. An algorithm (such as C4.5 or ID3) [Quinlan, 1986] is employed to process these examples and generate a tree-like structure that represents associations and relationships between various attributes and their values that form these examples. The tree-like structure that is generated by generalizing from the known examples is called a decision tree.

Examples of situations, cases or events that are used to build the decision tree are called *instances* which are derived by combining values from a variety of sensors. These sensors and their values are assigned *attribute-value pairs* where each *attribute* can have one of many possible *values*. For instance, if a photo-detector is employed as a sensor its assigned attribute may be called 'light' and its values might be {'dark', 'bright'}. The labels directly correspond to numeric values output by the sensor and in a sense are context abstractions as discussed in the section on context awareness. Table 1 below shows an illustration of a few instances (or situations) that are obtained by combining various sensor attributes and their values. The column on the right indicates the known response to each of those instances.

Instance	Sensor attributes and their values					Response (Play outside, Watch TV)
	Light {bright, dark, medium}	Time of day {morning, afternoon, evening, night}	Day {weekday, weekend}	Humidity {high, moderate, low}	Temperature {Numeric}	
1	Bright	Afternoon	Weekend	Low	88	Play outside
2	Bright	Afternoon	Weekend	Low	34	Watch TV

Table 1. Sensor Attributes, Values and Responses for Two Sample Situations.

The table shows two situations where one is suitable for playing outside and the other is suitable to stay inside. It may be noticed that all the attributes including the response (or class) attribute are discrete, except temperature which is numeric. Such tables provide training data for resultant decision trees which generalize based on the given examples and their responses. More information on decision trees may be found in [Quinlan, 1993] [Ryszard et al., 1998] [Mitchell, 1997].

Continuing the discussion on the use of decision trees for interruption awareness the following are some advantages tree based methods provide:

- Suitable for continuous as well as categorical data. It was seen in the discussion on context aware systems and user interruption that information obtained from sensors about users activities, situations, behavior and surroundings is highly variegated. Decisions trees are not concerned with the data type as long as the categories and responses are well known.
- As mentioned before input is highly varied and can be continuous, discrete, nominal or even numeric. Since decision trees are not concerned with regression, extrapolation or probabilities, it is easier to create and refine them without processing the input data too much (this is especially useful because context can be defined as labels).
- Decision tree learning is fast for considerably large datasets. Speed and accuracy are important in today's context-aware and pervasive environments due to the required responsiveness and dynamism. Classification is typically faster than model generation which is advantageous because once the model is generated it can be incorporated in the system and used for quick classification of new instances.
- Trees are easy to analyze and refine since the generated model is nothing but a set of 'if-then' like rules and policies. This format is easy to analyze both for the end-user as well as the designer.
- Suitable for medium to very large datasets without considerable decrease in speed and accuracy. Given today's scenarios, the number of attribute-value pairs is usually very large and decision trees are able to maintain speed and accuracy, while keeping a small model size.

- Generated trees can be easily employed to classify test data as long as the data formats are appropriate. Test data are previously unseen instances that are fed to the learned model by keeping the same format and attribute order. It is then possible for the model to classify the test data and predict whether it was correctly classified or not.
- Decision trees are robust and classify test data with a high degree of accuracy for a small number of training instances. In addition, trees are largely insensitive to missing attributes, instances or even responses and many techniques exist to increase the accuracy of trees in such cases to avoid over fitting problems.

In the following section, we will see how post-design preference elicitation methods can be used to refine the learned models in order to suit individual user preferences and modifications.

4.2.3. Post-design Preference Elicitation

The main approach of this research as seen in the previous sections is to employ information gathered from users in order to create interruption aware decision tree models which can then be used to interrupt the user appropriately. Although this approach takes user preferences and requirements into consideration it does so based on a statistical majority. In other words, if incorporated in a system it would be effective only to users who agree with the response majority. In order to satisfy the requirements of all users (including the minority) post-design preference elicitation has been proposed.

Research on preference elicitation has rapidly increased due to the rise in popularity of Internet based commerce. Before this, research was mainly focused on techniques for

personalization [Perugini et al., 2002], tailoring [Steimerling et al., 1997] and participation [Muller et al., 1993]. Personalization involved changing the configuration of user interfaces to suit individual users while tailoring involved modifying services, user interfaces and the underlying system model to suit individual or group user activities. Participation on the other hand entails active user involvement in the design or modification process of a system or an interface.

Preference elicitation is similar to the first two approaches since it involves modifying system functionality, interfaces or services. The difference lies in the fact that user input in the form of choices and preferences are used only as passive information to modify the learning model of the system. It is the learning model that decides when and how to incorporate the changes. Previous studies in preference elicitation were mostly manual and consigned to the field of decision support such as [Keeney et al., 1976] [Howard et al., 1984]. These involved the determination of preference functions based on factors such as relative weights, preference relationships or independence. Given the rising demand for preference based software tools and web based services, automating the task of preference elicitation was deemed essential in order to reduce the burden on designers and end-users. [Domshlak et al., 2001] and [Boutilier et al., 2004] employ graphical CP and TCP nets-based techniques for web based preference elicitation in order to qualitatively gather information about users' preferences while surfing web pages based on conditional dependence and preference relationships. The focus of their research is on examining how to perform preferential comparison of outcomes and how to find the optimal outcome given a set of preferences and preference relationships. Techniques from artificial intelligence and knowledge mining are also being employed to automate

the preference elicitation process. [Ha et al., 1998] employ case based reasoning techniques where new user preferences are classified and sorted based on similar historical cases. They also employ distance measures in order to determine similarity of new and previously seen cases. Other approaches [Lahie et al., 2004] use query based learning and valuation functions to elicit information from users and learn about preferences.

In this research, a simpler approach that makes use of exception tables is employed to test the effectiveness of post-design preference elicitation. Exception tables are cases of instances which have been flagged by the user as being unsuitable to their interruption preferences. Responses generated by the system are compared with entries in the exception tables, and based on users' new preferences, the responses are modified. This approach is suitable if the number of exceptions, i.e. responses not suitable to users is small. Once this number increases to a considerable size, comparison delays may be caused resulting in decreased efficiency of the system. Another disadvantage is the size of the table itself, which may be a factor in memory limited devices and systems. In such cases learning approaches may be used for preference elicitation instead of the static approach we employ. This discussion is presented in further detail in the section on future work.

CHAPTER 5

APPLICATION DOMAINS AND SCENARIOS

System generated interruption is a common occurrence in many devices and computing systems today. We noticed in the section on user interruption that interruptions generated by the system are very often essential and cannot be eliminated. It is their characteristics, timing and purpose that can make them beneficial or disruptive. In the following paragraphs, a few examples of popular application domains and scenarios are presented with some discussion about how system generated interruption can be an important issue in their design. We also look at the two scenarios that were selected for this research in further detail.

- Intelligent Tutoring Systems: [Brusilovsky et al., 1996] [Capuano et al., 2000]

Issues in this domain are mainly user-system interaction and user-user communication. Due to these frequent activities users' tasks and concentration are affected. It is essential to design tutoring systems that do not hinder individual user's activities because the main goal of such systems is training, learning and memory retention all of which can be affected due to interruption.

- Telephone, Mobile and Wireless Communication: [Hansson et al., 2001] [Marmasse et al., 2000] [Siewiorek et al., 2003]

Issues in this domain are mobility and its inherent relationship with different kinds of environmental variables. Interruption is one of the most common problems faced by wired and wireless phone users today. Given the complexity and variability of users' environments while using telephones and the related social and psychological considerations, interruption handling in this domain is very crucial.

- Office and Business Environments: [Dey et al., 2000] [Asthana et al., 1994] [Biegl, 2000] [Yan et al., 2000]

Issues in this domain are mainly related to busyness, noise-related interruptions, level and type of user activity and so on. Again, this domain is highly prone to interruptions by a variety of system and user (visitor in case of office assistants) generated events.

- Collaborative Environments: [Rauschert et al., 2002] [Cohen et al., 1997]

Issues in this domain are similar to office and business environments since collaborative systems are frequently used in such places for group work and design. In addition to interruptions generated by systems, user interaction and communication via the collaborative system is a common source of interruptions. With the growing reliance of businesses and organizations on such systems, interruption handling is very crucial.

- Chat and Instant Messaging: [Avrahami et al., 2004] [Campbell et al., 2002]

While relegated to the entertainment domain, chat and instant messaging is also used for group work and information sharing. Issues in this domain are frequent communications, notifications and so on. While these can be disabled, users' concentration and focus are reduced nonetheless in this scenario.

- Internet Push-pull Technologies: [Abowd et al., 1997] [Lockerd et al., 2003]

Issues in this domain are mainly specific to the manner of information exchange between system and users. Push based technologies force information onto users with little concern for timing and disruption causing interruption. Pull based technologies on the other hand force users to set aside their original activity and obtain information from the system. Effective system design [Norman, 1986] [Shneiderman, 1997] can assist in reducing hindrances caused due to such interactions.

- Wearable Computing: [Van Laerhoven et al., 2002] [Korteum et al., 1999]

Issues in this domain are mainly related to the modality of interruption since the devices and systems are body worn. Again, as in mobile and wireless domains, addressing issues regarding location, surrounding conditions, users' socio-psychological characteristics, etc. are essential for handling interruptions.

- Medical Device Monitoring: [Liu et al., 2004] [FitSense] [Polar] [BodyMedia]

Issues in this domain are similar to wearable systems due to the fact that many sensors are body worn. Other issues include noise sensitivity, distributed monitoring, and so on.

- Automated Command and Control: [McGee et al., 1998] [Cohen et al., 1997]

Issues in this domain are similar to the collaborative systems domain, and mobile and wireless domain. Most automated command and control systems are highly complex distributed collaborative systems involving notifications, updates, user-user communication and activities such as decision support, information sharing and so on. Interruption handling is essential due to user activities such as decision making, information awareness and situational responsiveness.

- Automated Transportation Systems: [Postema et al., 1998] [Rillings, 1997]

Issues in this domain are related to the modality and type of interruption due to the user's focus and concentration on driving the vehicle. Interruption handling in this domain involves the choice of active versus passive interruption modalities, timing and duration of interruptions.

We notice that every system above is an example of a highly responsive and dynamic environment that involves frequent user interaction, notification and communication. In other words, each generates some form of system generated interruption that is also essential to its functioning. For example, instant messaging software functions solely on the basis of interrupting users with incoming messages. While all these are possible candidates for the incorporation of interruption handling functionality we address and narrow down four scenarios, two of which are chosen for further analysis. These scenarios are then used for the development of the learning models and the interruption aware simulator.

The first scenario is a desktop PC and the second is an instant messaging scenario. The third is a context-aware cell/mobile phone and the fourth is a virtual secretary scenario. The cell phone and virtual secretary scenarios were chosen from the four selected scenarios for this research. The cell phone scenario was chosen due to its wide popularity and incessant interruption causing property. The virtual secretary scenario was also chosen because of the incessant interruption causing activities in office environments due to frequent visitors. All scenarios are very common today, and the nature of the services they provide necessitates interruption handling properties to be incorporated in their design. Below we discuss each scenario in brief.

5.1. Desktop PCs

From the list of possible application domains and scenarios, desktop PC usage is one of the more common scenarios. With the growing use of computers for a variety of applications, the desktop and mobile PC scenario is an important consideration due to its frequent interruption causing nature. The very nature and design of interfaces and services provided such as MS Windows, Mac OS, etc., causes interruptions that cannot be avoided because they are necessary for human computer interaction. Table 2 shows how such a scenario can be incorporated with interruption handling capabilities by employing a variety of sensors, attributes and contexts.

Use Cases (during interruption event)	Contexts						
	Duration Timestamp	Action- Listener	Process- Listener	Event- Listener	Global/Local		Busy? How to Interrupt?
User is in working	None	Keyboard	Winword	Dialog box	At screen		Yes, Flash Icon
User is watching movie	High	None	nvcdvd	Real-time virus scan	None		Yes, Queue
User is idle	High	None	winamp	Real-time virus scan	At screen		No, Dialog Box
User is sleeping	High	None	Winword	Windows update	None		No, Automatic

Table 2. Sample Use Cases and Possible Sensor Attributes for the PC Scenario

Below we list possible tools, sensors and situations to further develop such a scenario for interruption handling.

- Tools and sensors: desktop or mobile PC, typical operating system and software, video cameras, event listeners, loggers and monitors, action listeners, process (thread) listeners, web based monitors, light meters, microphone arrays, body worn sensors like temperature, heart rate detectors, etc.

- Input for interruption handling: system events, application events, keyboard and mouse events, all kinds of errors, log files, web history and preferences, user gaze and expressions (or presence), ambient lighting, speech and noise, physical state of user, and so on.

From Table 2 we notice that sensor data can be employed as attribute-value pairs to recognize various events and situations such as a user working on spreadsheets, watching movie, dozing and so on. In the larger context, this scenario can be easily applied to a smart home environment with the difference being that the sensors and enabling technologies are not limited to the PC but are spread across the home.

5.2. Instant Messaging

Table 3 shows another scenario that is already very popular for quick and easy communication, namely instant messaging (IM). IM can be highly disruptive in work environments but if used effectively can assist tremendously in group work and information sharing.

Use Cases (during incoming message)	Contexts						
	Duration Timeout	Action- Listener	Presence- Listener	Message Listener	Game/Use		Busy? How to Interrupt?
User is in working	None	Keyboard	Winward	Work	At screen		Yes, Flash
User is watching movie	High	None	winward	Friend	None		Yes, Email
User is idle	High	None	winward	Boss	At screen		No, Popup
User is sleeping	High	None	Winward	Colleague	None		Yes, Email

Table 3. Sample Use Cases and Possible Sensor Attributes for the Instant Messaging Scenario

Table 3 shows different use cases, sensor attributes and their values for the instant messaging scenario. This scenario is essentially similar to the desktop PC scenario because the core computing environment is the PC. The differences lie in the contexts of use and some sensor attributes such as the identity of the chatter, or the content and purpose of chatter's message. Below we look at the two scenarios that were chosen for the purposes of this research.

5.3. Cell Phone Scenario

The total number of cell phone subscribers was poised to top 2 billion by the end of 2005 [Lonergan]. This was a remarkable statistic since it means that roughly one in every three persons on the planet would possess a cell phone by 2005. In reality it indicates an extremely high concentration of cell phone users in places like large cities, public places, businesses and organizations. There have been many studies conducted to study the spread of cell phone use in human society [Castells et al., 2004] and their effect on human behavior, sociology and health [Bautsch et al., 2001] [Castells et al., 2004] [Palen et al., 2000] [Ling et al., 2003]. These efforts have studied the effect cell phones have on users' day to day activities, behavior, social interactions with other users, effect on tasks and health related issues. The general consensus is that cell phone interruptions are here to stay and the only way users can avoid interrupting effects is by following implicit rules or 'phone etiquette'.

Scientists and researchers are working on techniques that will enable cell phones to be aware of the users' interruptability thus allowing users to continue using their cell phones without having to be considerate to anyone around. Some efforts in this regard are

[Siewiorek et al., 2003] [Pedersen et al., 2001] [Milewshi et al., 2000] [Khalil et al., 2005]. Some of these techniques focus on providing cues to callers about users' contexts and situations while other methods focus on employing fixed state-based interruption handling using different modalities for different states. Other studies use static rules and policies to interrupt users based on calendar information, time, schedules, etc.

The approach proposed by this dissertation is unique in the sense that information gathered before and after design is *actively* used in learning about the interruptability of the user and allowing the phone to respond in an appropriate manner. In order to do so, surveys, field studies and experience sampling techniques were employed and will be discussed in the sections that follow. These activities provide information that enabled us to select attributes and their values for creating interruption-aware learning models which recognize new instances based on the information provided by users. This process is also discussed later in the dissertation.

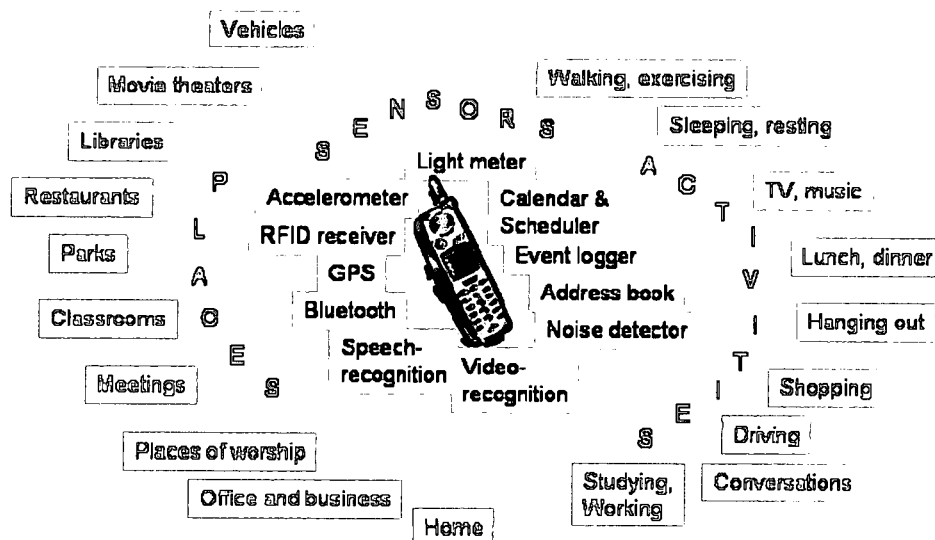


Figure 9. Cell Phone Scenario with Possible Sensors and User Contexts

Figure 9 illustrates a possible cell phone scenario with a variety of sensors residing on the phone and in users' surroundings, and different types of user contexts such as activities, places and other situations. This research chooses such a cell phone scenario wherein the user performs day to day activities, and is in a variety of situations and surroundings. She may or may not have her cell phone with her, and her phone may be switched on or off. In other words, it is a generic scenario similar to a real life scenario (assumptions will be discussed in the section on scenario development). The cell phone itself is assumed to have a variety of sensors for detecting user contexts and interruptability, and based on the learning model it responds to every situation in an appropriate manner. The user environment on the other hand is deliberately proposed to be sensor free in order to prevent intrusiveness and pervasiveness of technology.

Use Cases (during an incoming call)	Contexts							Response
	Voice Data	Ambient Noise	Phone Motion	Time	Phone Usage	Calendar Schedule		Busy? How to Interrupt?
User is in meeting	Presence of Voices	None	No	Day	None	Scheduled Meeting		Yes, Flash icon
User is in conversation	Presence of Voices	None / Medium / High	None	Day / Night	None	None		Yes/No, Beep, Vibrate
User is jogging	Absence of Voices	Medium	High	Day	None	None		No, Ring
User is sleeping	Absence of Voice	None	No	Night	None	None		Yes, LED, screen msg.
.								
.								
.								

Table 4. Sample Situations, Context Attributes and Interruptability Responses in Cell Phone Scenario

Table 4 illustrates a sample set of situations a user is in, their corresponding recognition attributes, and the preferred response of an interruption-aware cell phone system. As the table shows, users can perform a variety of activities while keeping their

cell phones with them such as going to meetings, conversing with other users, exercising, sleeping and so on. In the section on scenario development we will see a general summary of users' day to day activities and situations. Many of these activities can be represented by a combination of context-attributes such as voice data, ambient noise, phone motion, time, usage, etc. These context-attributes in turn can be represented by actual sensors which gather data about these context attributes and their values. An interruption aware cell phone may respond to these activities in case of impending interruptions by performing any of the listed tasks such as flash icons, beep, ring, flash LED, etc.

5.4. Virtual Secretary Scenario

Office environments are also a very common source of interruptions. This is especially true in the case of environments with frequent visitors. Assistants and secretaries have traditionally been the means for interruption handling for users in such environments. Over the past few decades a few attempts have been made to automate these assistants by developing systems that mediate between users and visitors and handle many of users' common office tasks.

Most attempts however are static rule and policy based [Gruen et al., 1999] [Gardner, 1981] [Dabbish et al., 2003]. Some of these rely on heuristics predefined by designers for performing activities and mediating between users and interruption events. Some efforts focus on interruption handling by employing context and situation awareness [Yan et al., 2000] [Mitchell et al., 1994]. These use some form of machine learning technique such as neural networks or decision trees in order to learn about users'

contexts from external sensors. Some efforts are multimodal and interactive in nature [Yankelovich et al., 1996] and focus on interaction via animations or speech. The focus of such systems is more on user friendliness and human computer interaction than on interruption handling. These studies suggest that the focus is generally on automating office based activities rather than on actual interruption handling itself, and given the growing amount of research on CSCW, and automated systems, it is necessary to incorporate interruption handling as a property in itself.

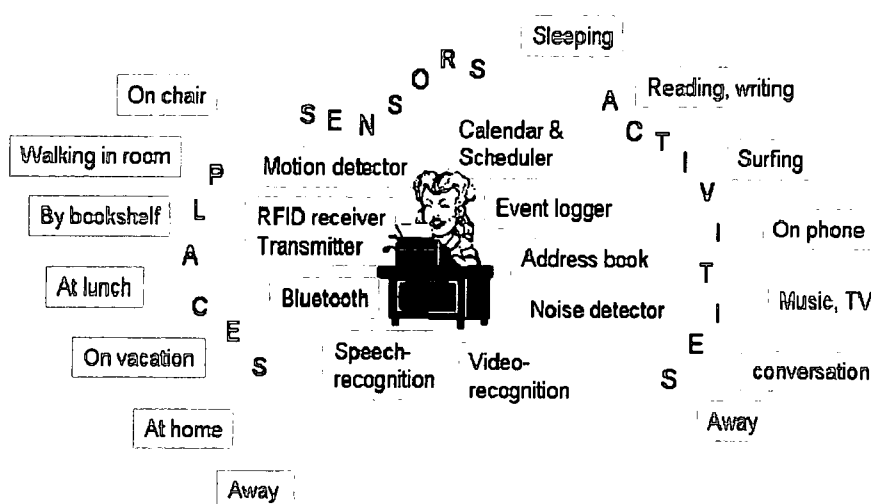


Figure 10. Virtual Secretary Scenario with Sensors, User Activities and Locations

Our approach to interruption handling in an office scenario involves the presence of a virtual secretary as a mediator between the user and any incoming visitors. Such a mediator is essentially a device connected to a variety of sensors both in the user's office and in the visitors seating area. The mediator gathers information from these sensors to determine user's availability and interruptability based on learning mechanisms, and then responds to an incoming visitor appropriately. The user may be performing a variety of

activities such as reading, writing, surfing the web, chatting online or on the phone and so on. She may also be at a variety of locations in and out of the office based on which the virtual secretary may respond to incoming visitors. The job of the virtual secretary for our purposes is not to manage user activities and assist her in performing her tasks, but simply to perform interruption handling when visitors arrive and take messages from them. Thus in this scenario, the virtual secretary may ask the visitors to walk into the office if the user is not busy, to leave a message, check back later or to wait in case user is not in or is busy.

Use Cases (when person wants to meet)	Contexts							Response
	Voice Data	Pressure Data	App. Book	Time	Phone Usage	User Info.		Busy? How to Interrupt?
User is already in meeting	Presence of Voices	Chairs occupied	Yes / No	Office hours	None	Scheduled Meeting		Yes, LED
User is on the phone	Presence of Voice	Chair occupied	Yes / No	Work hours	Yes	Colleague		Yes, LED
User is working	Absence of Voices	Chair occupied	Yes	Work hours	None	Student		No, beep
User is at lunch	Absence of Voices	Chair empty	Yes / No	Lunch hours	None	Student		Yes, Answering Machine
.								
.								
.								

Table 5. Sample Situations, Context Attributes and Interruptability Responses in Virtual Secretary Scenario

Table 5 shows the above discussion in the form of different types of situations or use cases, their recognition contexts as seen by sensors and possible responses of the virtual secretary. For example if the user is in a meeting, the situation could be recognized by voice data, chair pressure data, appointment schedule or time. Once the virtual secretary recognizes these contexts based on a variety of sensors, it could respond

in different ways such as to ask the visitor to leave a message. It could then interrupt the user in the office based on her interruptability, in this case via an LED. The details of this scenario and its development based on sensor and attribute selection, etc. will be discussed in the section on scenario development. In the following chapter we look at initial design and analysis of the cell phone and virtual secretary simulators.

CHAPTER 6

INITIAL DESIGN AND ANALYSIS

In the previous section we elaborated on various scenarios that involve system generated interruption and narrowed our focus to two commonly occurring scenarios. We also briefly expanded the selected scenarios discussing sample use cases, situations, contexts and possible sensor alternatives. This section focuses on the initial design steps for the design of simulators for these scenarios in order to validate the interruption handling approaches and ideas presented in this research. Here we will look at the design methodology that is proposed as a part of this work for the design of interruption aware systems. The discussion here focuses on the use of surveys, field studies and experience sampling for data collection and training of the simulator model.

This study follows a methodology that involves an active incorporation of user information from various techniques such as surveys, field studies and experience sampling for the design of interruption aware systems.

In the following sections we elaborate this proposed methodology and discuss its various stages in detail.

6.1. Design Methodology

Engineering design processes generally rely on a standard set of stages that need to be performed in sequence starting with activities like requirements analysis, goal and needs determination and so on, and ending with prototyping and evaluation [White et al., 1976] [Ertas et al., 1993] [Middendorf, 1989]. Many stages are repetitive because a developed system, model or prototype needs to be refined and modified to suit design goals and requirements. Software design processes are similar to engineering design processes (for example the Waterfall Model, or the Spiral Model), with the main differences being the actual system design stages which may differ based on design approaches, application domains and enabling technologies [Behforooz et al., 1996].

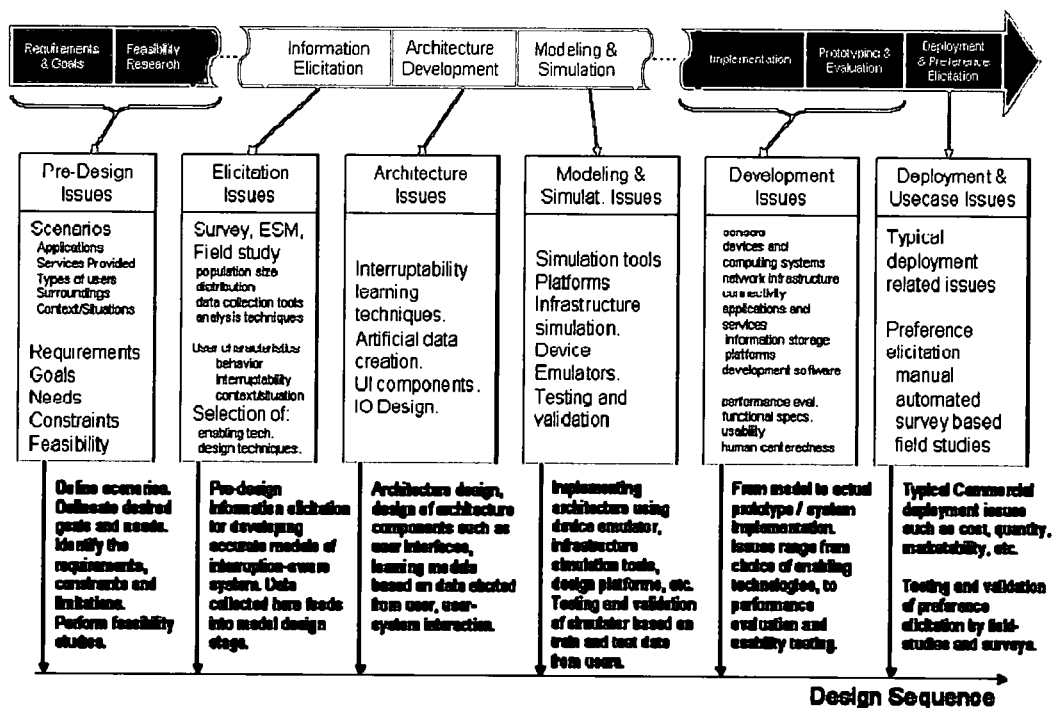


Figure 11. Methodology for the Design of Interruption-Aware Systems

For the purposes of this research, a methodology based on a typical software design process is introduced. Figure 11 shows the design process that incorporates this methodology. The idea behind this methodology is to employ information collected from users directly towards the selection, modeling and design of various components in an interruption aware system. This idea was discussed in detail in the section on the proposed solution.

As the figure suggests the beginning and end stages of the design process are similar to a conventional software design process. Prior to the actual design and implementation phases, scenario definitions, requirements analysis, goals, objectives and needs definitions and feasibility studies are performed. For example, to design an interruption aware cell phone designers need to formulate the scenario completely, and ascertain its limitations, constraints, use cases and environments. They also need to determine the functional requirements of such a system based on objectives, goals, and needs from various user studies. Feasibility studies need to be performed to determine whether interruption aware cell phones can be used in the given conditions and situations, what their projected development costs might be, and other factors that would limit their feasibility.

Once all pre-design stages have been completed the methodology proposed here involves a pre-design information elicitation stage to gather data for use in the actual design process. This is done by employing a variety of user based data collection techniques like surveys, field studies and experience sampling. Currently a manual data collection process is used but other approaches similar to user based design, participatory design, etc. can also be employed to automate the information elicitation process. This

information elicitation stage provides designers with knowledge about selection of attributes, sensors, and other enabling technologies for the implementation and prototyping stages. Other issues such as user preferences, characteristics, surroundings and interruptability are also ascertained here to be incorporated into the system design.

The information obtained is then used in the actual architecture design, modeling and simulation stage which is nothing but the initial design stage of a typical software design process. Here the data is incorporated into learning algorithms to generate models for interruption awareness. These models are then integrated with simulator tools (e.g. device emulators in case of our cell phone scenario) and software modeling and development tools like Java to create working simulators for the scenarios. Issues that are addressed in these stages are the choice of interruptability models (e.g. in this dissertation decision tree learning models are employed), training data, rules and policies, architecture components, user and component interfaces, etc. Modeling and simulation issues are addressed once architecture development is complete. These issues may be related to the choice of simulation tools, platforms, interface development tools, device emulators and other software or hardware components.

Developed models and simulations are tested and validated for functional accuracy. Once this is done the development phase can continue with system prototyping and implementation using a host of enabling technologies, the choice of which normally has been made during the initial design and elicitation stages. Prototypes developed are evaluated and tested not only for functional performance, but also usability, ergonomics, and other human and system factors such as user-friendliness, human centeredness, robustness, and security.

Our methodology adds a post-design preference elicitation stage in the design process (discussed in the section on proposed solution). In other words, after the interruption aware system has been deployed in the real world, users can perform preference based modifications to system functionality by providing it required information directly.

6.2. Information Elicitation

The following sections present in detail the initial stages of the design process, namely the pre-design information elicitation, scenario development, and data integration. The information elicitation stage is divided into three parts based on the techniques that were used in this work. The scenario development is based on the information that is gathered from this elicitation stage. The data integration stage essentially uses the data and results obtained from information elicitation to develop learning models for the design of the interruption-aware simulators for the two scenarios considered here.

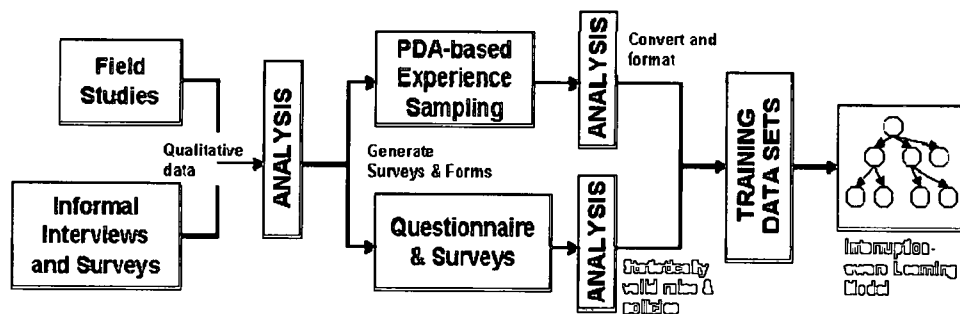


Figure 12. Incorporating Elicited Information to Build Interruption-Aware Models

Figure 12 illustrates the process of information elicitation in detail. Three information elicitation methods from human factors and usability engineering were chosen, namely field studies, experience sampling, and user surveys. Field studies,

informal interviews and surveys provide valuable qualitative data for designers to proceed with the actual information elicitation stage. Upon analysis of this data, designers can create surveys and forms for PDA-based or survey based elicitation methods. Each of these methods provides different kinds of data, and that data is eventually converted to training datasets by formatting, or conversion from statistically valid rules and policies. These training datasets are then used for creating interruption aware models which can then be employed to respond to interruptions in an appropriate manner. In this section we look at the three techniques for information elicitation, namely field studies, experience sampling, and user surveys.

6.2.1. Field Studies

Field studies are one of the most popular and highly result-oriented methods for gathering comprehensive information about intended users and scenarios for consideration before designing any product. In addition to providing data about users, they also provide designers with valuable data about use cases, constraints and limitations, alternative design approaches and preferences for enabling technologies. This is essential because it cannot be achieved as effectively by other methods such as surveys or interviews. Such studies allow designers to visualize the proposed system and its functioning by observing its users and its intended deployment environments. They also allow designers to develop the scenarios further by elaborating on issues such as design assumptions, constraints, choices and functionality.

For the cell phone and virtual secretary scenarios, field studies were conducted in order to ascertain information about two main components.

- Environments: to determine in what physical environments the proposed systems would be used. In case of the cell phone scenario, the environments were highly variable and included indoors, outdoors, high-noise, low-noise, and private and public spaces. In case of the virtual secretary scenario, the environments were static and limited to an office room and a visitor lounge. This made the scenario development structured and bounded by imposing limitations on environments of use, and hence the choice of enabling technologies and other design essentials.
- End-users: to determine relationships between users and their environments, the nature of activities they perform, their behavior, responses and preferences to various events and situations. For the cell phone scenario, end-user issues are very important because these phones can be used by a variety of users. Important factors to be considered here are user preferences (response to interruptions) in different situations, activity, characteristics, psychological issues, and group behavior. For the virtual secretary scenario such factors are not very important since the system caters to individual users. Factors that are essential however are busyness levels, work styles and schedules, interruption preferences, visitor priorities, and so on.

There are many ways of conducting effective field studies such as direct in-field observation, video-audio recording of user activities and situations in real-time scenarios, interviewing users, use of time diaries, and indirect monitoring. For this research, direct on-site observation was done to gather qualitative data about users and their environments for both scenarios. Informal interviews were also conducted with cell phone users and with secretaries about their activities, needs, and environmental specifications. The observations that were made and the results of the informal interviews

enabled the creation of a comprehensive list of activities performed by users, and places in which users can be located in both the cell phone and virtual secretary scenarios.

VIRTUAL SECRETARY SCENARIO FIELD STUDY INFORMATION	
Cater to professor's needs	
Assist students of the department	
Menial office activities (answer phones, mail, supplies, etc.)	
Chat, gossip and non-work related activities	
Projects provided by professor	
Paperwork, records and academic documentation	
Visitor management and gate keeping activity	
Tools used: desktop computer, intercom, phones,	
Software: instant messaging, work related software (office, SPSS, etc.)	
General sequence of activities in case of incoming visitors:	
Welcome visitor	
Ask for the visitors purpose of visit	
Obtain information about the visitor	
Check if the professor is in her office	
Perform various activities based on situation	
Inform professor about visitor	
Take message and contact info. from visitor	
Respond on behalf of professor	
Request to be seated	
Deny permission to visit	
Activities performed in case of incoming visitor are generally based on:	
Purpose of visit	
Whether or not visitor has an appointment	
Identity of the visitor	
Professors activities and busyness level	
When professor is away, she is usually:	
In meetings	
At lunch	
In conversation with other professors in the building	
Across campus	
On vacation, at conferences, or abroad	
When professor is in, she is usually:	
Busy working on papers, course work, conferences, etc.	
Conversation with students, colleagues, etc.	
Not busy (surfing the web, reading, etc.)	

Table 6. Information Obtained from Direct Observations and Informal Interviews of Departmental Secretaries in University Professors' Offices.

Table 6 shows a summary of the information obtained from direct observations and informal interviews of four departmental secretaries in university professors' offices. The table provides information about the activities of the professor, the secretary and the

locations of the professor when she is away. Table 7 shows a similar summary of interviews and field studies with ten users who own cell phones.

CELL PHONE SCENARIO FIELD STUDY INFORMATION
List of possible user activities: Office/work related Meetings, presentations Classrooms, teaching, studying, etc. Site visits, tours, etc. Interviews, conversations, discussions Movement based work Seated, PC based work Manual labor Vehicle related Driving vehicles As passenger in vehicles Airplanes Home related Sleeping, resting Watching TV, listening to music, reading Working on car, lawn, etc. Hanging out with friends and family Lunch and dinners Shower and bathroom Other Eating out at fine restaurants or fast food places Exercising outdoors or in gym Hanging out in parks, malls, streets In cinema theaters, opera houses, concerts
Types of calls Personal and business (work) related
Types of callers Family, friends, work related
Purpose of calls Casual (no particular intent), emergency, intended
Response to interruption Don't care Mild annoyance Frustration Anger
Use the cell phone for the following activities Make and receive phone calls Listen to music and watch videos Surf the web Schedule events, look at calendar, play games, SMS
Employ ringer, beeps, music, vibration for notification
Possible phone locations are Shirt, jeans pockets, belt, nearby desk, purse, bookbag

Table 7. Information Obtained from Informal Interviews of Cell Phone Users.

Direct observation was not possible due to the high variability in user locations throughout the day. Therefore users were interviewed and provided answers about their day-to-day activities while their cell phone was with them. They were also required to recollect how they responded in case of incoming calls, what their situation during that time was, what notification modes their phones were set to, and so on. This information was vital in understanding the two scenarios and designing the simulations accordingly by keeping the rules, constraints and preferences in mind. It also enabled the development of effective user surveys required for building the learning models for interruption awareness.

It can be seen that field studies provide valuable information about a variety of design parameters such as environmental constraints, user activities and preferences. While field studies are important for gathering preliminary design information, they can be very informal and qualitative in nature and the data they generate is not directly usable in the design process. This is especially disadvantageous because the approach employed here aims to directly use information gathered from users in developing the system. Therefore to aid this 'user driven' approach, other methods such as experience sampling and user surveys are employed for direct data collection.

6.2.2. Experience Sampling

Another effective method of gathering information about users and environments for the given scenarios is the Experience Sampling Method (ESM) proposed by Larson and Csikszentmihalyi [Larson et al., 1983]. The idea behind this method is to obtain a direct user perspective and response to situations or events occurring during her day-to-

day activities in a given environment. In other words designers try to find out what the user was doing when a given event occurred, what her surrounding conditions were and what her response to the event was. By gathering information about such topics using small forms or surveys, designers have a clearer idea when they incorporate human system interaction functionality in the proposed system design. Experience sampling also helps designers gain a better understanding about users' socio-psychological behavior, characteristics and responses to events, timeliness, synchronization, activity sequences, and interruptability.

For this research the PDA based experience sampling technique was employed for two purposes listed as follows:

- To gather qualitative information about users' activities, situations, contexts, environments, behavior and preferences.
- To employ the PDA based experience sampling tool to collect instance data for developing learning models for interruption awareness.

Conducting an experience sampling study usually requires users to complete forms or surveys repeatedly. This process is controlled by making the user fill out the form at random times, periodically, or upon the occurrence of events. Thus, data about users' responses, situations, behavior, etc. is gathered multiple times or for every occurring event. Experience sampling can be carried out using paper and pencil methods or by employing software and computer based methods. A good review of experience sampling and its techniques is conducted by Conner [Conner et al., 2001].

Computer and software based techniques include web based, PDA based, voice and video recorder based, and so on. The least automated of these techniques require

users to carry electronic forms or diaries and some kind of interruption device that enables them to sample their experiences periodically. Of these, PDA based techniques have become popular due to the flexibility and ease of use they provide [Intille et al., 2003]. PDA based techniques require the surveys or questionnaires to be downloaded onto the PDA and then distributed to multiple users. The PDA is then modified to interrupt the user periodically or at random and answer the survey repeatedly. The user can also answer the questions based on events and situations occurring in her environment. The surveys and questionnaires developed for PDA based techniques are usually multiple-choice based and elicit less detailed information from the user.

Computer and electronic device based methods are the easiest way to conduct experience sampling. Effective analysis of information can also be performed because all data is stored on the device and can be used directly by experts or software for analysis. One disadvantage these types of methods have is that the setup and infrastructure costs may be prohibitive for some studies, and since multiple devices may need to be distributed to users, the costs naturally go up, and there are also concerns of security and theft of devices. The other disadvantage they have is the user may not be able to enter information as freely as possible in paper and pencil based methods. This is one reason why computer and device based methods generally use multiple choice type surveys and forms.

Figure 13 illustrates how software based experience sampling can be employed for collecting training data for developing interruption aware learning models for the simulators.

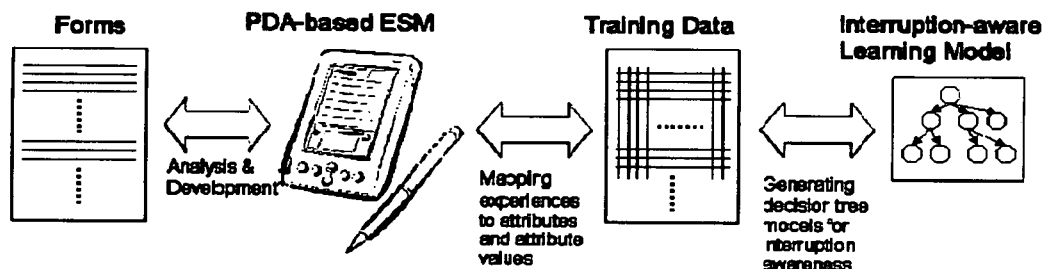


Figure 13. PDA-Based Experience Sampling for Generating Interruption-Aware Learning Models

As the figure shows, special forms or surveys are generated that can be installed on the PDA. These forms are a set of multiple choice questions that gather information about users' response, context, surrounding situation and preferences in case of incoming interruptions. Figure 14 shows a sample PDA based form. This is actually generated on a PDA simulator but similar forms have been installed on actual PDA's and tested.

Figure 14. Sample PDA Based Experience Sampling Form

The forms shown in figure 13 allow users to enter a variety of information that is stored on the device. It is then transferred to a desktop PC where it is converted to

formatted training datasets (a sample is shown in Figure 14) for the machine learning algorithms (in our case the decision tree algorithms) which then generate the interruption aware models. The models are subsequently used to respond to new situations and events.

PDA Model: Palm Tungsten E
Operating System: Palm OS v. 5x
Programming Language: J2ME Wireless Toolkit
Platform: Windows XP
Number of Experiments: 16
Number of Users: 6
Average Samples Per Day: 18
Number of Instances: Virtual Secretary: 102 Cell Phone: 0

Table 8: Summary of PDA Based Experience Sampling

Table 8 is a summary of the PDA based experience sampling study performed. As shown, experience sampling yields a very small number of samples or instances required for creating learning models. These disadvantages are discussed later.

```
@relation firstsetresponded
@attribute onphone {yes, no}
@attribute motion {yes, no}
@attribute chairpressure {yes, no}
@attribute soundtype {speech, music, noise, none}
@attribute presence {yes, no}
@data
no,yes,yes,music,yes
no,yes,no,music,yes
no,no,yes,music,yes
no,no,no,music,no
no,yes,yes,noise,yes
no,yes,no,noise,yes
```

The data shown above represents a sample training dataset generated by analyzing and converting PDA based experience sampling data into an appropriate format recognizable by the learning algorithms.

While experience sampling (both pencil and computer based) has many advantages for qualitative information elicitation it is not very effective for collecting actual training data for the generation of machine learning models because:

- It does not generate an extensive set of training data. Our experiments generated about 20 instances in a typical 8 hour work day. In order to generate effective learning models with a large number of attributes and values it is necessary to have a large dataset to start with.
- The required dataset size for minimal accuracy increases drastically as the number of attributes and attribute values grow. At the same time, the number of instances (or experiences) sampled remains approximately the same since the frequency of sampling is usually independent of the contents of the training data.
- Smaller training datasets generated using experience sampling cannot cover all possible instances and use cases which are not incorporated in the learning model. This makes the developed system more prone to misclassifications and incorrect responses to interruptions. This is partly due to the highly routine nature of human activity.
- Another flaw of this technique is that rare and exceptional cases may be completely missed in the generated data. This might happen because human activity is highly routine and rare cases and exceptions are not encountered on a daily basis.
- Forms that elicit information need to be accurate enough to represent the attributes and attribute values in the training data. If they are not, it creates an overload on the experts who analyze and convert the data into training datasets.

- Periodically sampling for experiences using PDA's itself causes interruption to the users because now every time an interruption occurs they have to stop their activities and record the interruption and the ensuing responses in the PDA.

Some advantages of employing PDA based data collection are that they are real-time data collection tools. The users' record data immediately after an event or interruption has occurred. In this manner there is no loss of information and all situations, events and responses are captured effectively. This is useful during the initial design stages because there are no sensors to collect training data for modeling and simulation purposes. In the following sections we will look at the main information elicitation technique used namely the user surveys where we discuss how surveys can be employed to generate effective training data for the interruption aware learning models.

6.2.3. User Surveys

The illustration in figure 12 described the process of information elicitation starting with field studies and informal surveys and ending with the learned model for interruption awareness. In this section we present a third technique for information elicitation, namely user surveys and questionnaires.

Surveys and questionnaires have long been a mode of data collection and information elicitation for designers attempting to gain an accurate understanding of the scenarios under consideration. Excellent literature on survey design research can be found in [Alreck et al., 1995] [Oppenheim, 1992]. They provide a variety of statistical and preference based information that can be used effectively to make design choices. In

performing this research, surveys and questionnaires were used as a primary means for gathering quantitative data on the following user characteristics:

- Activities
- Interruptability
- Busyness levels
- Preferences
- Locations
- Response
- Surrounding environments

Surveys and questionnaires provide many advantages over field studies and experience sampling methods such as:

- They provide a first hand account of users' preferences, their activities, response and other characteristics that may not be perceivable through observations or interviews. Observations may not yield enough information while interviews may miss certain issues because they are designed by the interviewee or a usability expert rather than the end-user herself.
- They provide information that is normally quantitative in nature. Even if the information is qualitative (such as phrases, labels, or nominal values), it can be effectively categorized, weighted and converted to numbers for easy analysis.
- Analysis of data resulting from surveys and questionnaires allows the effective creation of concepts, rules and policies that are statistically valid. In other words the generated rules correspond to actual responses from the user population and can be used directly for system design.

- They are able to provide an understanding of the relationships between various design components and issues such as activities, places, preferences, responses and so on. Such relationships may not be observable via field studies, interviews or experience sampling techniques.
- Many of the design goals are determined directly from survey and questionnaire analysis. Since they represent the end-user perspective on how the eventual system design should behave, they are an accurate representation of design goals, needs and requirements.
- They also enable designers to foresee shortcomings in design approaches or existing system designs and hence can be employed in post-design or deployment stages also.

While surveys provide many advantages they have one inherent drawback. They do not take into account the real-time behavior of users, systems or environments, and hence provide no valuable data about realistic functioning and behavior of these entities.

In the following sections, a detailed discussion of the surveys used for the cell phone and virtual secretary scenarios is presented along with analysis of these surveys. We then look at how the results obtained from this analysis can be used to generate training datasets and interruption aware learning models.

6.3. Information Analysis

In order to develop the cell phone and virtual secretary scenarios four surveys were created. Each of these surveys elicited a variety of information about the users ranging from their day-to-day activities to their preferred responses to interruptions.

Figure 15 shows a sample survey designed for the cell phone scenario. The actual surveys are similar in nature, and are attached as appendices A, B, C and D.

I Cell Phone Related Questions

1. What kind of mobile devices do you use? List name and model number.

a. Cell phone [_____]

b. Handheld PDA [_____]

c. Smart phone [_____]

d. None

2. What activities do you perform using your mobile device?

a. Make and receive phone calls.

b. Surf the web (check email, etc.).

c. Send and receive SMS messages.

d. Store and update information (addresses, numbers, emails, etc.)

e. Perform calendaring and scheduling (make schedules, reminders, notes, etc.)

f. Document-related activities (word, powerpoint, excel, etc.)

3. What kind of notification modes does the device have?

a. Voice

b. Beeps

c. Vibration

d. LED (light)

e. Other (screen display, flashing icons on screen, animation, music, etc.) [_____]

Figure 15. Sample Questionnaire for the Cell Phone Scenario

The surveys elicit different kinds of information in the form of multiple choice answers, free text and paragraphs, numerical choices and answers in the form of two-dimensional tables. Later in the section we will see how this information is converted to statistically applicable cases that can be then used for generating rules and policies for the interruption aware learning models. The following is a summary of the surveys in terms of the queries that were posed to the respondents for both the virtual secretary and cell phone scenarios.

- a. Ascertain the different types of notification modes on users' cell phones. It is important to know the options (or the lack thereof) users have in terms of notification

modes. This is necessary to validate the need for interruption handling and situation awareness for cell phones.

- b. Ascertain the number and types of default notification modes frequently used by cell phone users. Compare this with the available notification modes. This may indicate that people generally have multiple choices but do not switch notification modes unless there are exceptional cases. This may also indicate the need for automated interruption handling and notification to ease user workload and prevent them from having to switch modes from time to time.
- c. Ascertain information about user privacy issues and preferences when it comes to having intrusive technologies in their daily environments. Also ascertain whether users prefer having personal information broadcast in the public domain. This may be essential in the selection of sensors and other enabling technologies like RFID, OTA-based networking, etc.
- d. Ascertain whether users would like to know in advance details about the caller (such as her number, name, purpose, etc.) in order to allow them to decide whether to answer the phone or not. This is also a privacy issue since callees may be interested in such information while the callers may not want to divulge it (especially in case of telemarketers, sales persons, etc.).
- e. Ascertain user preferences in situations that make it difficult for them to use their default and preferred notification modes (such as high noise, or noise sensitive locations). If such situations are frequent, then it is essential that interruption handling facilities be incorporated in the phone design to prevent users from having to manually change settings (and occasionally forget and cause interruptions).

- f. Ascertain user priorities when it comes to answering phone calls and being interrupted based on four main factors, namely timeliness, situation, purpose, and caller. Any of these factors may override users original preferences, or notification modes and an interruption handling mechanism must be able to incorporate these subtle priorities.
- g. Ascertain preferences of users when it comes to different types of users such as families, friends, acquaintances, colleagues and unknowns. Generally family may take the first priority but this might be true only when the user is not at work. Knowledge of such relationships between various parties can also enhance the interruption handling mechanism and make it more accurate.
- h. Ascertain what users do with cell phones that have been left by other users. This may shed light on their own preferences when forgetting their cell phones at various locations. In some situations it may be wise for the interruption handling mechanism to lock the phone for security and privacy reasons while in other situations it may be prudent to leave it unlocked in case of emergencies.
- i. Ascertain the different types of activities users perform with their cell phones. Given the state of technology today there are different kinds of devices ranging from basic cell phones to smart phones integrated with PDAs and media players. It is necessary to assess the complexity and nature of services that such devices provide so that in case of incoming calls, the interruption handling mechanisms can appropriately notify the user causing the least amount of interruption.
- j. Ascertain the proximity of cell phones to their users during their daily activities. This is necessary to understand how to interrupt the user with respect to other factors such

as location, noise sensitivity and situation. In some cases, the high proximity of the phones to the users (such as on a belt clip, etc.) may benefit interruption handling by preventing the use of loud auditory notification modes.

- k. Ascertain the different types of activities performed by users daily, and the locations in which users perform these activities. While our research does not focus on the recognition of user activities this information is necessary to determine the relationships between user preferences, types of notification modes to be used, locations, noise sensitivity, proximity of phones, and so on. This information is also essential in determining the choice of sensors and other enabling technologies.
- l. Ascertain the different types of activities performed by student assistants and department secretaries. Again, our research does not focus on the recognition of activities but only on the determination of how interruptible the professor in the office is and how the virtual secretary should respond to visitors.
- m. Ascertain the different types of devices used by student assistants and department secretaries to perform their tasks. This information is necessary to determine how they perform their tasks, communicate with the professor and respond to incoming visitors. Some secretaries may rely entirely on telephones and intercom while others may prefer face-to-face communication. This information would help in determining the choice of enabling technologies to be used.
- n. Ascertain the number of visitors, types of visitors and the purpose of visitors arriving at the department offices each day. The number of visitors provides information about the frequency of interruptions and can help designers to control the interruption handling with respect to timing and closeness to each other. Information about the

types of visitors (such as students, colleagues, unknowns, etc.) enables designers to build interruption handling in relation to priority and importance. Information about the purpose of their visit (such as casual drop-ins, appointments, emergencies, etc.) allows designers to incorporate interruption handling with respect to user preferences and importance of the visit.

- o. Ascertain the sequence of activities performed by department secretaries and student assistants in each case of an incoming visitor. This is helpful to designers in automating some of that functionality for interruption handling.
- p. Ascertain the response of secretaries and professors (users inside offices) to frequent incoming visitors. Such responses may vary from mild annoyances to anger and frustration. This information is also helpful in the cell phone scenario because it helps in choosing notification modes for interrupting the users.

In the sections below we present results of the surveys distributed to cell phone users, department secretaries and student assistants at the University of Dayton.

Cell phones today come with a variety of notification modes ranging from the basic ringer and music, to voice and animated avatars. Table 9 describes a list of possible notification modes for cell phones, PDAs and smart phones.

<i>Possible notification modes and their levels for the cell phone scenario</i>
Ringer – low, medium, high volume (short, medium, long duration)
Music – low, medium, high volume (short, medium, long duration)
Vibration – low, medium, high frequency (short, medium, long duration)
Beeps – low, medium, high volume (short, medium, long duration)
Text – static, moving text (basic or backlit)
LED (light) – one level (standard duration)
Animations – avatars, flashing icons (color, backlit, etc.)
Voice – low, medium, high volume (standard duration)

Table 9. List of Notification Modes and Different Types of Levels for Mobile Devices

Based on the field study and survey results we noticed that all the users possessed the basic ringer and vibrate notification modes (figure 16). LED, music and text were the next most common available modes while voice and animations were very rare. This is probably because voice and animations are currently limited to high-end phones such as Smartphones which are also expensive. The sample size for this response was 66, and the users were university students, faculty and staff with considerable cell phone use experience. This response indicates that there are already relevant choices available for notification modes for interruption handling in cell phones and other kinds of mobile devices.

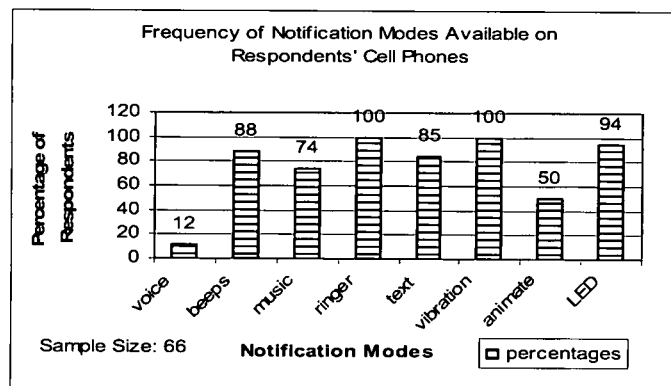


Figure 16. Percentage Distribution of Notification Modes on Respondents' Cell Phones

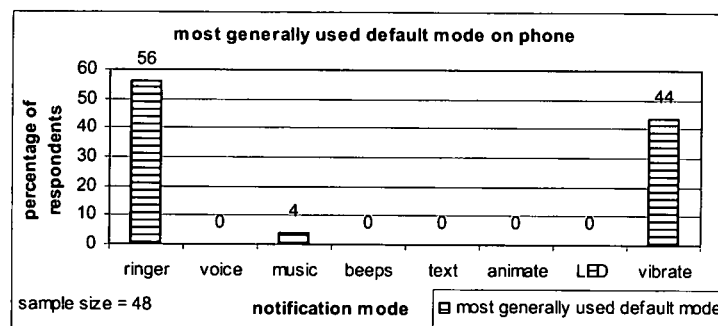


Figure 17. Percentage Distribution of Default Notification Modes Used by Cell Phone Owners

When asked about the default notification modes on their cell phones, i.e. the mode that they use most of the time, it turned out that the basic notification modes, namely ringer and vibrate are the most commonly used notification modes (figure 17). Very few respondents preferred to use music as their default notification mode.

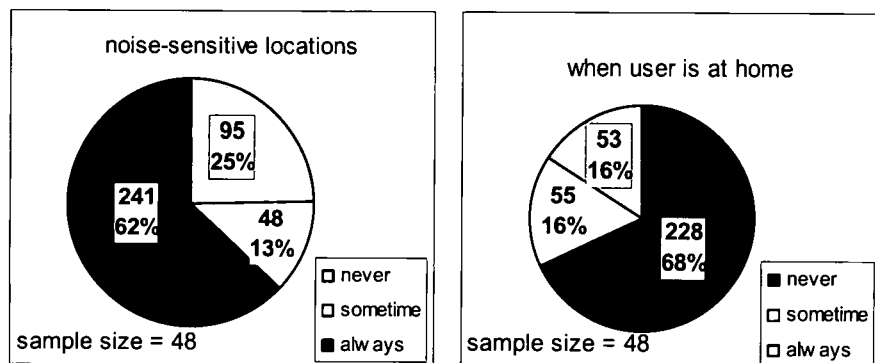


Figure 18. Percentage Distribution of Users Changing their Default Notification Modes for Noise Sensitive and at Home Locations

Figure 18 shows that a majority of users (68 percent) do not change their default notification modes when they are at home while when they are at other noise sensitive locations this number drops down to 25 percent. These numbers were obtained by querying users about their preferences during various activities at home and also when they are at various noise sensitive locations outside their homes. These queries were combined to yield a summary for noise sensitive and at home instances which are shown in the charts above. This finding is important because it will indicate that users are not generally likely to change their notification modes except in case of situations such as emergencies or noise sensitive locations. This response validates the need for interruption handling functionality in cell phones and automating the change of notification modes based on situations.

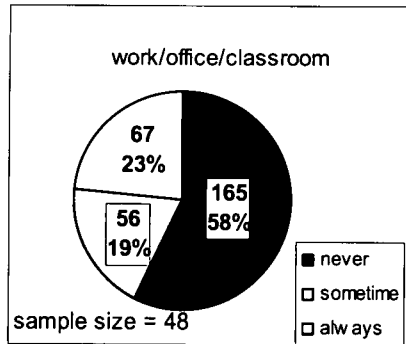


Figure 19. Percentage Distribution of Users Changing their Default Notification Modes when in Work Related Locations

Figure 19 shows that a majority of users (58 percent) never change their settings even in work related locations such as offices, classrooms meetings, and so on. This indicates a high likelihood that users in these locations will be frequently interrupted by incoming calls.

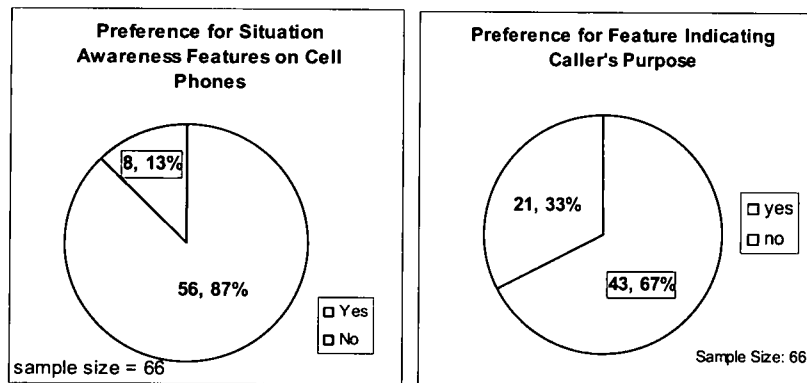


Figure 20. Preferences for Situation Awareness Functionality, and Features Indicating Caller's Purpose in Cell Phones

End-user privacy and security are important design criteria in distributed environments today. While not a design-altering issue, it is nonetheless a factor in case of the cell phone and virtual secretary scenarios. Respondents were asked about their

preferences for situation awareness and access to personal information by incorporating sensors in their daily lives. Figure 20 suggests that a large majority of users (87 percent) would prefer such features but there are also a non-negligible number (13 percent) that would not want their personal information broadcast and their phones to know about every situation and location they are in. Furthermore, when asked whether they would like to have features indicating caller information such as purpose on their cell phones (second chart in figure 20), we see that the numbers drop down to 67 percent in favor and 33 percent against. This indicates that users generally do not want their personal information to be broadcast to other phones but at the same time would like to know more about the callers themselves before they pick up the phone.

Many devices today already come with features that indicate the callee of the caller's information (currently restricted to entities like phone number, name and time). To our knowledge there are no commercial cell phones that indicate the callers location or purpose. Given the preferences of users towards the inclusion of such features it is predicted that future devices will provide users with more information about callers.

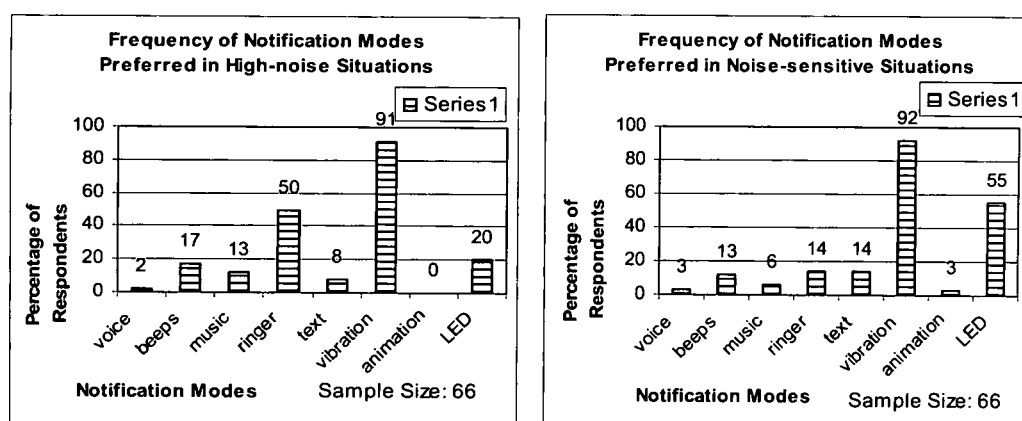


Figure 21. Preference of Notification Modes in High-Noise and Noise Sensitive Situations

From Figure 21, we notice that users generally prefer to use vibration or ringer for notification purposes when they are situated in high-noise locations. While in noise sensitive situations the preference for ringer, music and other active interruption modalities goes down as can also be seen in figure 21. Another interesting fact is that users prefer to use active notification modes for interruption in case of emergencies related to work, family or friends. This is irrespective of their situation and location. This can be noticed in figure 22 where the combined active modes such as ringer, music and beeps outnumber the combined passive modes such as vibration, text, animations and LED by a large percentage.

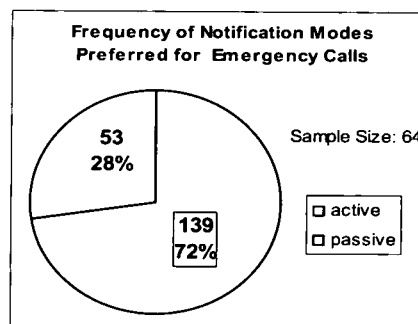


Figure 22. Frequency of Combined Active and Passive Notification Modes in Case of Emergencies Irrespective of Location and Situation.

When it comes to deciding whether to answer phone calls (Figure 23) users may choose based on a variety of options such as the type of caller, purpose, situation or time. In cases where users are at home it turns out that they prefer to choose based on one of these options rather than a 'don't care' option. The 'don't care' option is more frequently used when users are outside in public places like malls and theaters. Users are very likely to decide whether to answer based on caller, purpose, situation and time when there driving or in vehicles as passengers.

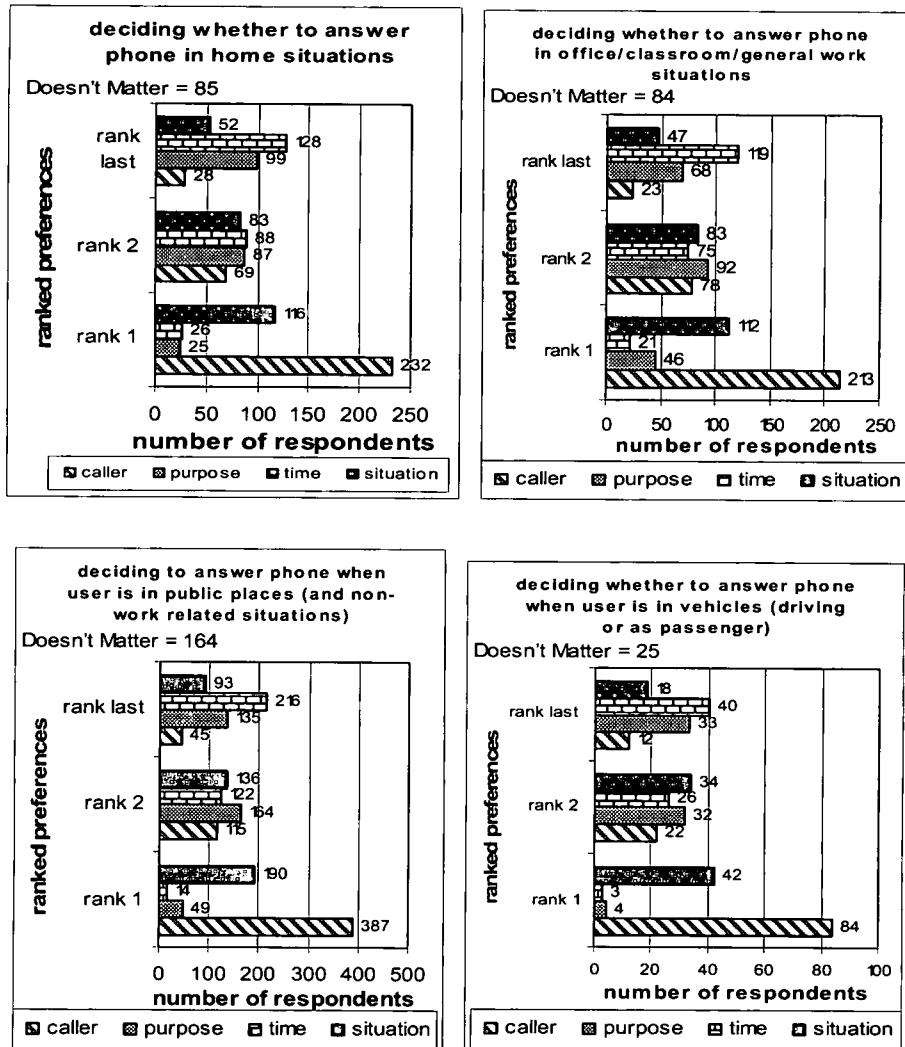


Figure 23. Deciding Whether to Answer Based on Four Choices in Home Situations, Work Related Situations, Public Places, and in Vehicles

As the charts suggest users prefer knowing who the caller is first before deciding whether to answer the phone or not. Caller is consistently ranked as the first priority in making this decision. The second rank is evenly distributed as users choose to look for the caller's purpose or situation when answering the phone. We also notice that users

generally don't look at these options when they are at in public places but are more likely to do so when at home or in vehicles.

The fact that caller is used as the first decision maker by users when deciding whether to answer incoming calls is further validated by the following charts.

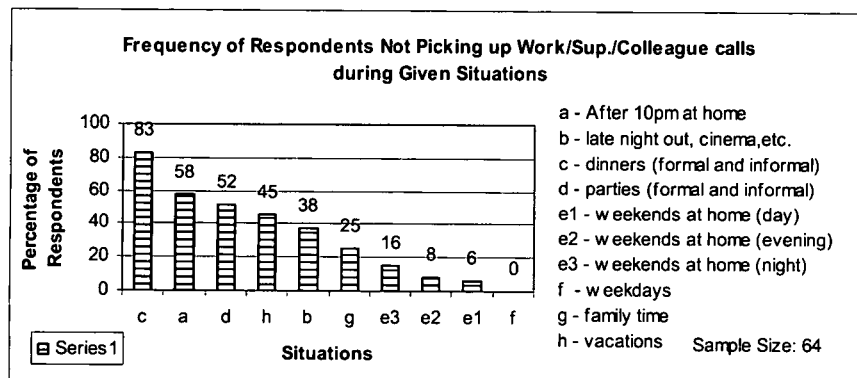


Figure 24. Percentage of Work Related Calls Picked Up Based on User Situation

In figure 24 we see that when users are at home or on their personal time with family, friends, on vacations or parties, they generally do not pick up phone calls from work or work related callers.

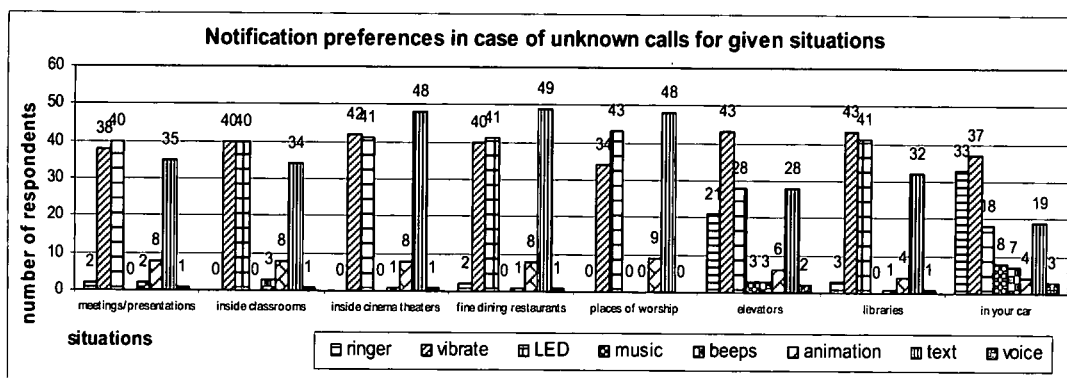


Figure 25. Number of Respondents Preferring Given Notification Modes for Various Situations in Case of Unknown Calls

In figure 25 we see that in case of unknown calls, users generally prefer to employ passive notification modes such as vibrate, LED or text for different kinds of situations. This changes, as seen in figure 26 when the caller is a family member or friend and the purpose is an emergency where active notification modes like ringer are also preferred.

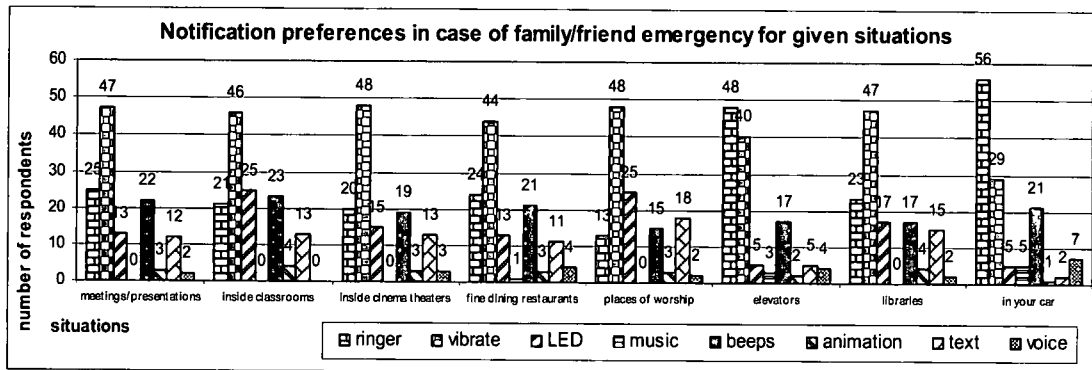


Figure 26. Notification Mode Preferences for Family or Friend Emergencies for Different Situations

When users forget their cell phones at various places they may either prefer the phone to lock itself or do nothing based on where the phone was forgotten. This is based on their own preference of answering other users' phones that were forgotten. Figure 27 indicates that users answer phones of family and friends more than they do of co-workers or unknown users.

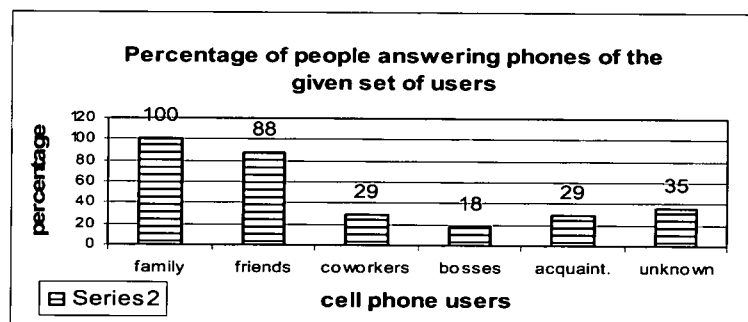


Figure 27. Percentage of Users Answering Other Users' Phones

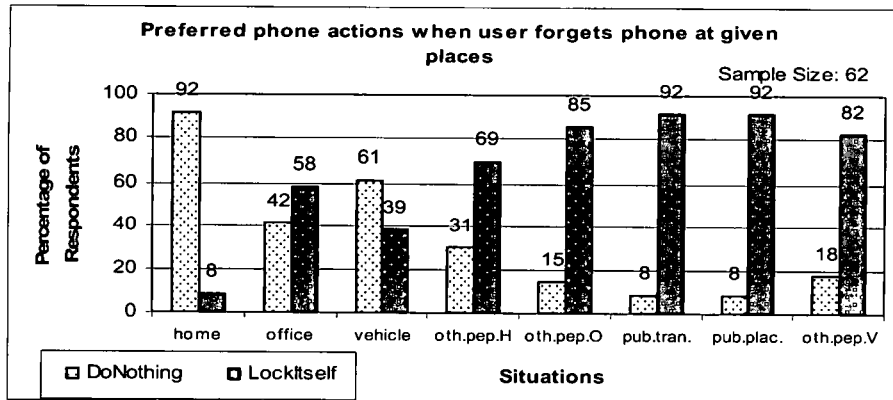


Figure 28. Preferred Phone Actions when User Forgets Phone at Given Locations

We notice from figure 28 that users generally prefer their phones to be locked when they are forgotten at other peoples' homes, public places or in vehicles.

In order to avoid interruption to surrounding users in case of nearby phones that have been forgotten, respondents were asked how they would prefer the phones to notify them in such cases. Figure 29 shows that users generally prefer passive modes when their phones have been forgotten at other peoples homes, offices or public places, while if they have been forgotten at home users prefer active notification modes.

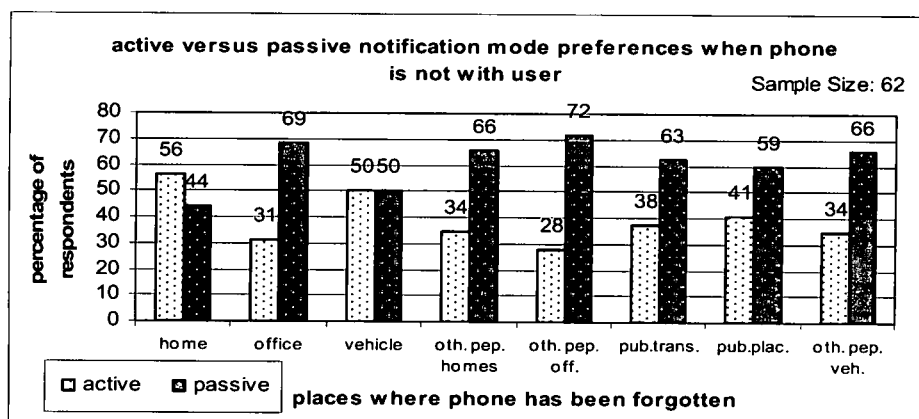


Figure 29. Preferences of Notification Modes when Phones Have Been Forgotten at the Given Places

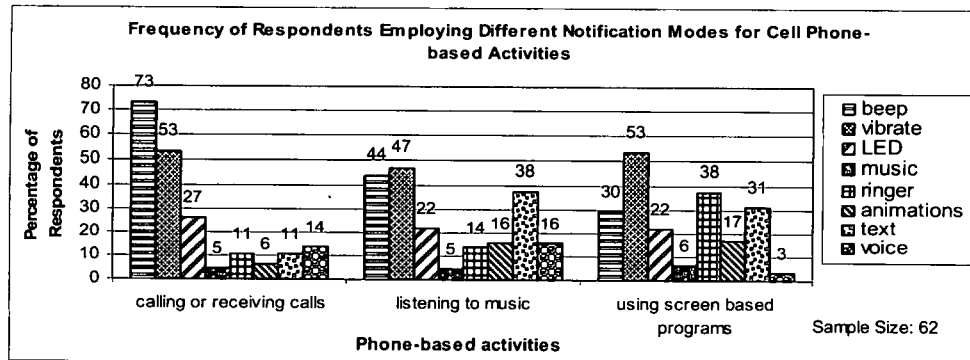


Figure 30. Percentage of Respondents Preferring Different Notification Modes when Performing Three Activities with their Cell Phones

Figure 30 shows what notification modes respondents prefer when they are performing three kinds of activities with their cell phones. The first activity is that of calling and receiving calls with phone held to the ear. In this case a majority of respondents prefer being notified by beeps or vibrate and LEDs. In case they are listening to music or watching videos on their phones with earphones on they prefer vibrations and beeps or even LEDs. We see here that the choice of music or ringer is low. When the respondents are using screen based programs like calendaring or scheduling tools, games, etc. they prefer to be notified mostly by vibration, ringer or beeps. Some users also prefer text due to the fact that they are already looking at the screen.

Users keep their phones at various locations throughout the day. In order to make it easier for designers to incorporate interruption handling functionality, it is necessary to ascertain where the phones are generally located. The first chart in Figure 31 illustrates a combined group of home locations such as bedrooms, living rooms, gardens and dinner tables. We see that users generally do not have their phones on their person, i.e. they are located nearby on desks, etc. The second chart illustrates combined work related

situations where users generally prefer to keep their phones with them (in their pockets, or purse). The remaining two charts show similar phone locations when users are in vehicles and public places (other than work related public places).

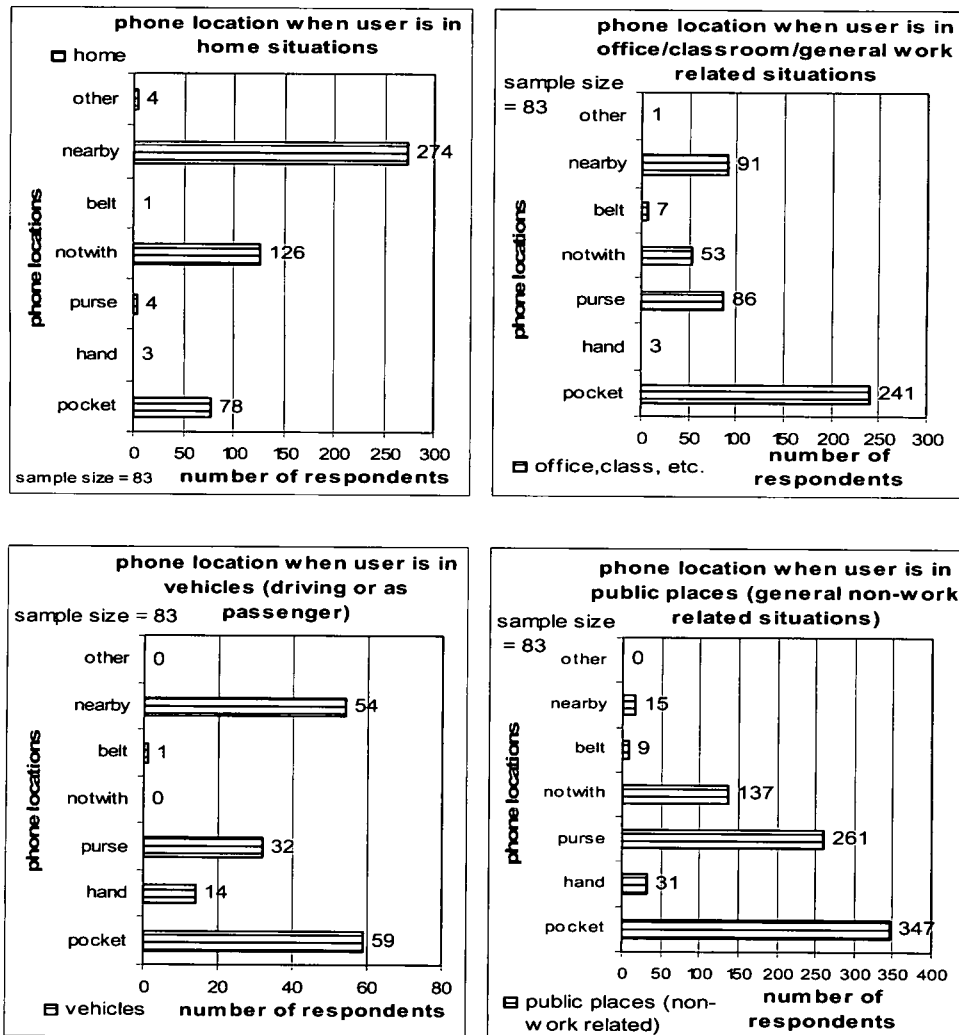


Figure 31. Phone Locations when User is in Home Based Situations, Work Related Situations, Public Places, and in Vehicles

We also notice when users are outside at work or in public places or even in vehicles, their phones are generally in their pockets, purses or nearby in book bags, etc. Such information can help designers in incorporating functionality related to notification

modes for specific situations. Table 10 below lists a summary of activities and places for respondents along with the most probable phone locations and preferred notification modes ranked from lowest to highest by vibrate, ringer and music respectively.

activities and places	phone loca	ringer	voice	music	beep	text	animate	LED	vibrate
shower, bathroom	notwith	51	11	37	17	7	1	15	33
walking on streets, parks	pocket	65	9	45	16	10	0	11	44
driving vehicle yourself	nearby	62	12	40	16	10	1	14	45
resting in the evening	nearby	64	6	43	16	11	1	16	45
trivial activities (TV, books, etc.)	nearby	65	8	45	17	11	2	15	45
outside movie theater	pocket/pur:	63	13	42	16	12	2	13	46
department stores and malls	pocket/pur:	63	13	42	16	12	2	13	46
cooking, cleaning, laundry, etc.	pocket/nea	63	9	39	14	8	1	17	47
in vehicle as passenger	pocket	62	11	45	18	13	1	13	48
sleeping at night	nearby	55	7	32	15	17	2	23	50
physical office work	pocket/not	36	8	20	8	12	2	18	52
indoor sports activities	notwith	36	7	20	12	12	2	22	55
outdoor sports activities	notwith	39	8	23	8	11	1	19	55
in school outside of classrooms	pocket	52	7	29	11	18	7	18	57
having dinner with family	pocket/not	34	1	23	14	15	6	28	58
travel based work	pocket	46	4	22	16	15	6	27	58
restaurants (TGIF, wendys, etc.)	pocket/pur:	49	7	32	12	14	2	19	59
friends, relatives places	pocket	56	5	36	16	10	3	19	59
not working while in office	pocket/nea	40	7	25	11	20	5	24	60
seated, PC based office work	pocket/nea	41	3	24	12	19	5	28	63
noisy public places	pocket	48	2	24	10	18	6	23	66
formal dinners and parties	pocket/pur:	21	0	13	6	26	9	34	67
work in noisy environments	pocket	40	2	21	10	19	6	23	67
inside movie theater during film	pocket	12	0	6	7	31	9	39	68

Table 10. Common Activities and Places of Respondents Along with Data About Most Probable Phone Locations and Preferred Notification Modes

The same kind of information can be summarized for noise-sensitive activities and places and we notice the change in preferred notification modes (Table 11).

activities and places	phone loca	ringer	voice	music	beeps	text	animatic	LED	vibrate
in your car	nearby	56	7	5	21	2	1	5	29
elevators	pocket/pur:	48	4	3	17	5	2	5	40
fine dining restaurants	pocket/pur:	24	4	1	21	11	3	13	44
inside classrooms	pocket/pur:	21	0	0	23	13	4	25	46
libraries	purse	23	2	0	17	15	4	17	47
meetings/presentations	pocket	25	2	0	22	12	3	13	47
places of worship	pocket/pur:	13	2	0	15	18	3	25	48
inside cinema theaters	pocket/pur:	20	3	0	19	13	3	15	48

Table 11. Common Noise-Sensitive Activities and Places of Respondents with Data About Most Probable Phone Locations and Preferred Notification Modes

We see that while users prefer active modes ring and music for a variety of activities their preferences for text, LED and vibrate modes increases as the noise sensitivity increases.

Results obtained from field studies and surveys for the virtual secretary scenario are now presented. Table 12 summarizes the activities performed.

<i>Cater to professor's needs – high priority</i>
Assist students of the department – high priority
Menial office activities (answer phones, mail, supplies, etc.) – medium priority
Chat, gossip and non-work related activities – none/lowest priority
Projects provided by professor – high priority
Paperwork, records and academic documentation – high priority
Visitor management and gate keeping activity – high priority
Welcome incoming visitors
Take messages in case of professor unavailability
Schedule appointments
Direct visitors to other places

Table 12. List of Activities Performed by a Student Assistant or Department Secretary

Here we notice that all activities except chat, gossip, lunch and other menial office activities such as check mail, supplies, etc. have highest priority. Since this research focuses mainly on gate keeping and interruption handling in case of incoming visitors, we divided visitor management into four categories: welcoming and responding to visitors, taking messages for visitors, scheduling appointments, and directing visitors. Of these four this scenario focuses on welcoming and responding to visitors only because this is where interruptions to the user inside the office can occur.

From the surveys we noticed that on an average there were 6-10 visitors coming in to see the professors. These numbers were larger during the start and end of the semesters due to the fact that more visitors came in for activities such as registrations,

funding and applications. The visitors can be categorized into four distinct types: students, colleagues, VIPs and unknown visitors as shown in Table 13.

Students: undergrad – 2 to 3, graduate – 5 to 6, PhD – 1 to 2
Colleagues (other professors) – 2 to 3
Dean (other V.I.P's) – 1 to 2
Unknown visitors – 1 to 2

Table 13. Average Number of Different Types of Visitors

From the field studies in table 7 we noticed that visitor information handling was based on four distinct priority categories, namely appointment, purpose of visit, visitor identity and professor's busyness level. Upon further analysis we noticed that these categories are directly linked to the type of visitors. In other words, colleagues and VIP's received higher visitation priority than say students or unknown visitors. While colleagues and VIP's always received highest priority (whether it was a casual drop-in, intended, or emergency), students and unknown visitors also received high priority in case of emergencies or if they had a scheduled appointment. In case the professor was busy, VIP's and colleagues received higher priority than students and unknown visitors.

It was also noticed that visitors are asked to wait if the professor is busy and the secretary has prior knowledge that the professor is not going to be busy for long. Visitors are also asked to wait if the professor has stepped out shortly such as gone to the bathroom or gone for lunch. However, this is true only in case of students and unknown visitors. In case of VIP's and colleagues the secretaries usually ask to leave a message, since it is assumed that the VIP's and colleagues are themselves busy people.

Lastly, student assistants and department secretaries check for users' presence by directly going inside their office to see if they are in or not. They also check for users' busyness levels by directly asking the users if they are busy and if they have time to see any drop-ins or colleagues, VIPs and unknown visitors. Designers can automate this process by installing multiple sensors for detecting the presence and busyness of users inside. Table 14 represents a list of possible responses secretaries use in order to respond to visitors. These responses can be employed by designers in developing virtual secretaries

<i>Possible system responses for a virtual secretary scenario</i>
Ask user to wait
Ask user to check back later
Allow user to walk into the office
Ask user to leave a message and contact information

Table 14. Secretary Responses to Incoming Visitors

In the following sections we develop the design further by concentrating on scenario development and design of the simulator.

6.4. Scenario Development

In the previous sections we looked at the definition of the virtual secretary and cell phone scenarios and also discussed the initial design phases of field studies, experience sampling and survey based design. Based on the results obtained from these techniques this section looks at the development and refinement of the two scenarios that are being considered.

6.4.1. Cell Phone

The main goal in the cell phone scenario is handling interruptions caused by incoming calls. In order to do this, we need to understand how the phone is being used and what the users' situations are when they have their phones with them. Below is a generalized list of all possible use cases for the cell phone scenario. The list deliberately avoids any specific activity or location due to the fact that activity recognition is not the goal of this research.

- There is no incoming call.
- There is an incoming call.
- The phone is with the user.
- The phone is near the user.
- The phone is far away from the user.
- The phone is being used.
- The phone is not being used, but is kept on.
- The phone is switched off.
- The user is at home.
- The user is in office, or other work related locations.
- The user is in a vehicle.
- The user is in other locations (not work related or home).
- The user is in a noise-sensitive location.
- Noise level around the user and her phone.
- Ambient light around the user and her phone.
- The caller is a family member, friend, colleague, or unknown.

- The time of day is morning, afternoon, evening, or night.
- Weekday or weekend.
- User is at rest or in motion.

For example, the user may be at home at night when the noise levels around her are low. This may indicate that she is sleeping or resting. In the same manner, the user may be in motion early in the morning on a weekend. This may indicate that she is exercising. Some of these use cases relate to user situations, activities and places, while others relate to her interruptability and to the systems response.

While these use cases are not adequate for recognizing specific activities, they are accurate enough to indicate the user's interruptability and to allow the system to determine how to interrupt her. Figure 32 illustrates a possible list of recognition attributes and their values that can be used to by the system to recognize the user's interruptability and respond appropriately.

Incoming Call	Yes	No		
Proximity	With User	Near User	Far Away From User	
Phone Usage	In Use	Not In Use		
Noise Sensitivity	Yes	No		
Location	Home	Office	Vehicle	Other
Caller Info.	Family	Friends	Work-related	Unknown
Caller Purpose	Emergency	Casual	Unknown	
Miscellaneous	Day of Week	Time of Day	Noise level	
	Video Image	Ambient light	Calendar Info.	

Figure 32. List of Possible Attributes and their Values for the Cell Phone Scenario

Thus the system could employ location sensors, calendar and date programs to determine what the user's situation was and interrupt her accordingly. Other sensors like noise and motion detectors could refine the notification modes being used depending on the situation. Based on survey results it was concluded that users do not mind having intelligent devices and systems pervading their daily environments assuming there are no privacy or security issues. Keeping this mind, our cell phone scenario tries to select sensors that would reside primarily on the phone rather than in the user's environment. The Table 15 below illustrates the set of sensor attributes and their values chosen for designing the cell phone simulator.

Sensor Attributes	Attribute Values
decibel	lessthan20db, 20to55db, 55to85db, morethan85db
soundtype	speech, noise, music
motion	high, medium, low, none
time	day, evening, night
usage	call, music, screen, none
light	bright, medium, dark
day	weekday, weekend
appointment	yes, no
caller	family, friend, work, unknown, none
context	casual, intended, emergency, unknown
location	office, home, vehicle, other
proximity	lessthan3ft, 3to20ft, morethan20ft
sensitivity	yes, no, notapplicable
response	ringer, voice, beep, LED, lockphone, donothing, vibrate, text

Table 15. List of Sensor Attributes and their Values for the Cell Phone Scenario

It is evident that most of the sensors are phone based except a few such as location, proximity and sensitivity. These three sensor attributes were necessary to simplify the situation recognition process and the user's interruptability. The sensor attribute appointment is not employed since it was found that users generally do not

adhere to their appointment schedules, and time delays or changes may affect the situation recognition process. Of these sensor attributes caller, context, location, proximity and sensitivity are directly used to detect the user's situation and interruptability, while the other attributes such as decibel level, motion, usage, etc. are employed primarily to refine the notification modes that are used for interruption handling. The decibel values were chosen from the NPC libraries Protective Noise Levels document [EPA Levels Document]. The last attribute is the class attribute. The model selects one of the given values as a response for any occurring situations or instances. It can be noticed that three of the responses are active interruption modes (ringer, voice, beep) and five are passive modes (LED, vibrate, text, donothing, lockphone)

Video and image sensor attributes were not used because they are impractical in a cell phone scenario. Certain assumptions are also made regarding the selection of some attributes. For example, it is assumed that the recognition software can easily differentiate between noise, speech and music. It is also assumed that the caller's purpose is always known except in the case when the caller is unknown. The other assumptions are related to the location of sensors. It is assumed that the user carries an RFID transmitter that lets the phone know about the user's proximity. It is also assumed that noise sensitive locations and other locations such as homes, offices and vehicle are installed with transmitters that let the phone know where it is situated.

Once these sensor attributes and their values were defined, the questionnaire results were again employed to generate training data for the interruption aware learning models that were used in the cell phone scenario. We will discuss this in further detail in the section on training data collection.

6.4.2. Virtual Secretary

Just as in the case of the cell phone scenario, the virtual secretary scenario also avoids specific activity recognition and instead focuses on the busyness and interruptability of the user in her office. This is required for the system to be able to interrupt the user appropriately and also let the visitor know whether she can be allowed into the office or not. The following list describes a set of use case attributes that were expanded on for the design of the virtual secretary scenario. Such generic use case attributes can be used by designers to select sensor attributes and their values for recognition activities.

- There is no incoming visitor.
- There is an incoming visitor (single or multiple).
- The professor is in the office.
- The professor is out of the office.
- Possible out of office locations are lunch, across campus, sick, meeting, etc.
- The professor is out of town.
- Possible 'away' locations are vacation, conference, etc.
- The professor is on the phone (in conversation or waiting).
- The professor is not on the phone but is busy.
- The professor is not busy.

As an example, if there is an incoming visitor the system may determine if the professor (or user) is available or not based on a variety of sensors situated in her office. If she is in the system may then determine if she is busy or not and then respond to the

visitor appropriately. Figure 33 illustrates a possible list of sensor recognition attributes and their values.

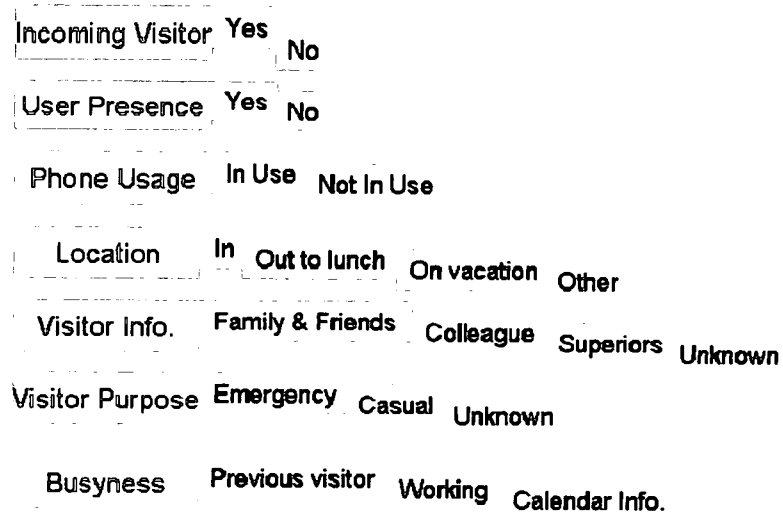


Figure 33. List of Possible Attributes and their Values for the Virtual Secretary Scenario

Sensor Attributes	Attribute Values
onphone	yes, no
motion	yes, no
chairpressure	yes, no
soundtype	speech, music, noise, none
presence	yes, no
busyattribute	set, reset
calendar	away, in
newvisitor	student, colleague, vip, other, none
appointment	yes, no
purpose	appointment, dropin, emergency, none
currentvisitor	student, colleague, vip, other, none
busytime	zero, 10min, 11to30min, morethan30min
awaytime	zero, 10min, 11to30min, morethan30min
allowvisit	allow, checklater, takemssg, wait, donothing

Table 16. List of Sensor Attributes and their Values for the Virtual Secretary Scenario

These attributes can be used to select appropriate sensors and other enabling technologies for the virtual secretary scenario. For example video cameras and motion detectors can be used to detect presence while speech recognition, calendar information, etc. can be employed for detecting the busyness level of the user inside the office. For the virtual secretary scenario the selected list of sensor attributes and their values is listed in Table 16.

The virtual secretary scenario is designed in a slightly different manner than the cell phone scenario. In the cell phone scenario, all the attributes are used to generate a training dataset that is used to build the interruption aware learning model. In the case of the virtual secretary scenario, the first four attributes: onphone, motion, chairpressure, and soundtype are employed to create a learning model that determines whether the user is in the office or not. The presence attribute is nothing but a class attribute. This model is further used along with the remaining attributes to build a second learning model that is actually employed to determine the interruptability of the user. This is illustrated clearly in the Figure 34.

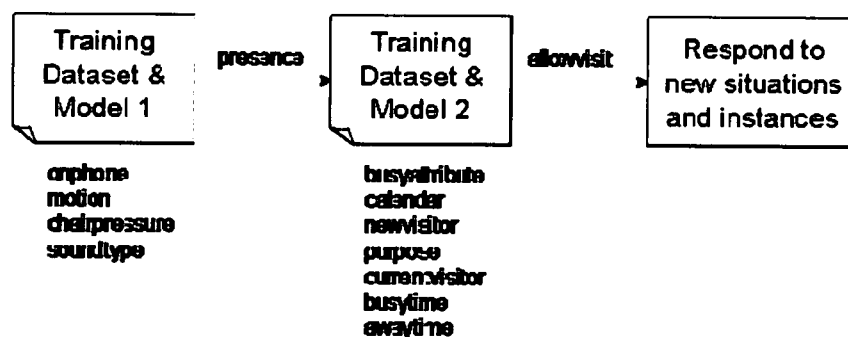


Figure 34. Dual Model Approach for the Virtual Secretary Scenario.

The advantage of the dual model approach is that it reduces the total number of attributes required to create the learning model and hence reduces the computational cost in terms of processing power, time and so on. In the next section we develop the two scenarios further by focusing on the next stage of the design process, that of the training data collection and model creation.

6.5. Training Data Collection

Normally training data is collected directly by employing a variety of sensors and data collection techniques. But in our approach, training data collection is the initial stage of the simulator design. At this point, sensors and other enabling technologies have not yet been finalized, and using sensor based data collection at this stage may result in models that are inaccurate. Inaccurate models would then be converted into full implementations by which time, it is too late to change sensors or other initial design choices.

In our approach all of the information is elicited from field studies and surveys and is used directly in developing generic rules and policies for the creation of the training datasets. These datasets are then employed to build decision tree models which look at new interruption cases (instances) and respond appropriately. The datasets created in this manner are not realistic since the data itself was not collected in real-time. The advantage this approach provides however is that user preferences and other characteristics of the scenario are accurately reflected in the training data. The other advantage is that many rare and exceptional cases can also be readily incorporated. Advantages and drawbacks of our approach to training data collection are discussed in

the sections on Information Elicitation, Field Studies and User Surveys. Below we discuss one example where we see how sample instances are converted into classified training datasets by incorporating rules and policies obtained from survey information. Table 17 shows a sample set of instances (or example use cases). The circled attribute values indicate the major differences between individual instances.

ATTRIBUTE VARIABLES AND THEIR VALUES												
C.No.	decibel	soundtype	motion	time	usage	light	day	caller	context	location	proximity	sensitivity
1a	less than 20db	noise	medium	day	none	dark	weekday	none	unknown	office	less than 3ft	no
1b	less than 20db	noise	medium	day	none	dark	weekday	none	unknown	office	more than 30ft	no
2a	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	office	more than 30ft	yes
2b	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	home	more than 30ft	no
2c	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	office	more than 30ft	no

Table 17. Sample Set of Instances Before Being Classified for Training Data Creation

Instances 1a and 1b indicate that there is no incoming call while the other three indicate that the caller is a family member. The context is also different and the family call is an emergency call. Some locations are office based locations while one is at home. The proximity of the phone to the user is also different. In one case it is with the user, while for the other four cases it is far away from the user. Also the noise sensitivity is set in one occasion when the user is in her office. The surveys and field studies provide the following information to create a training dataset out of this set of sample instances.

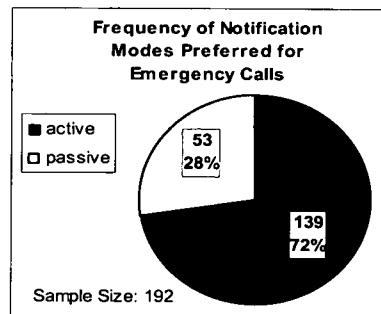


Figure 35. Notification Mode Preferences in Case of Emergency Calls

Figure 35 indicates that 72 percent of the users prefer active notification modes in case of emergency calls, i.e. they prefer to be interrupted with either voice, ringer, music, or beeps when there is an emergency call while Figure 36 indicates general user preferences for notification modes when the phone is not with them.

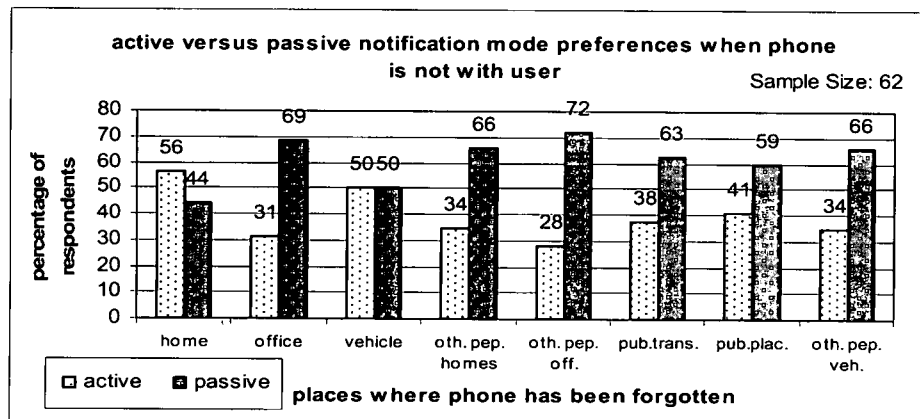


Figure 36. Notification Mode Preferences when Phone is Not with the Users

This indicates that when users are not at home, they generally prefer to be notified by passive modes except in cases of emergencies during which they prefer active notification modes. Therefore, we incorporate this in the form of rules or policies as follows:

```

proximity: morethan30ft
context: emergency
  sensitivity: yes → LED
              no, notapplicable
                location: home, vehicle, other → voice
                    office → LED
  
```

Noise-sensitivity is also a deciding factor in the above case. We know from the surveys that users generally prefer employing passive notification modes in noise

sensitive locations. So, even though there is an emergency, the designer has to realize that the locations may be noise sensitive and given the fact that the phone is not with the user, it makes sense to employ a passive notification mode.

CNo.	ATTRIBUTE VARIABLES AND THEIR VALUES												CLASS
	decibel	soundtype	motion	time	usage	light	day	caller	context	location	proximity	sensitivity	response
1a	less than 20db	noise	medium	day	none	dark	weekday	none	unknown	office	less than 30ft	no	do nothing
1b	less than 20db	noise	medium	day	none	dark	weekday	none	unknown	office	more than 30ft	no	lock phone
2a	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	office	more than 30ft	yes	LED
2b	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	home	more than 30ft	no	voice
2c	less than 20db	noise	medium	day	none	dark	weekday	family	emergency	office	more than 30ft	no	LED

Table 18. Sample Training Dataset Created by Integrating Survey Information with Sample Instances

Table 18 shows the sample training dataset created by integrating the rules and policies from the survey information with the set of sample instances in Table 17. We notice that when there is no incoming call the phone does nothing. But when there is no incoming call and when the phone is far away from the user, the phone locks itself depending on where it is situated. In this case, in instance 1b, the phone is situated in the office. We also notice that when the phone has been left in the office and there is a family emergency, it responds using a passive notification mode when the noise-sensitivity is set (maybe there is a meeting going on nearby), and responds using an active mode when the noise sensitivity is not set. For the complete list of rules and policies employed for creating the training datasets for both the virtual secretary and the cell phone scenarios, please refer to Appendices E and F.

Once the training datasets have been created the model and simulator design can proceed. In the next chapter we will discuss the architecture for simulation along with the tools and enabling technologies employed for simulating the scenarios.

CHAPTER 7

MODELING AND SIMULATION OF SCENARIOS

In this section we will look at the modeling and simulation of the two interruption aware scenarios selected for this study; the cell phone scenario and the virtual secretary scenario. Modeling and simulation is the next stage in our human centric design process. It enables designers to build software models of the intended system to ascertain whether the design choices that were made earlier are appropriate for the scenario and goals in consideration. It also allows designers to test the performance of the system and address issues such as time, cost and speed. These factors in turn affect other implementation choices such as enabling technologies and infrastructure. In this chapter we will look at various techniques and tools that were employed in developing these simulators although we first look at the architecture for the simulators in the section below.

7.1. Architecture for Simulation

For developing the architecture we employ a top-down approach wherein we first develop a generic architecture for an interruption aware system and then focus on each component in the architecture proceeding to the refinement of the architectural details.

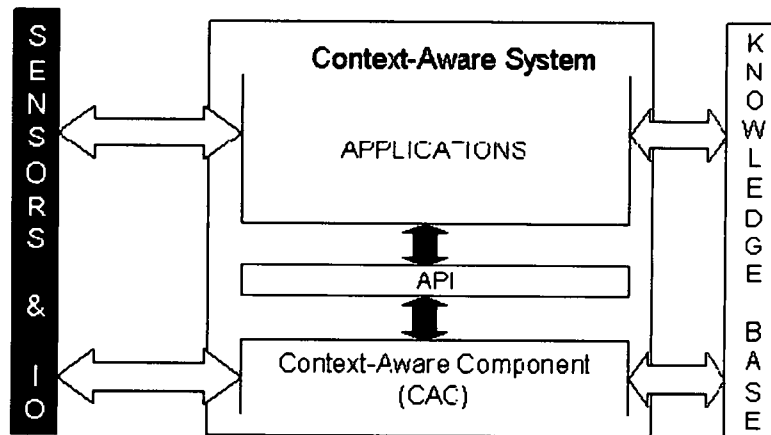


Figure 37. Generic Architecture of Context-Aware System

Figure 37 shows such a generic architecture for a context-aware system. Interruption awareness can be thought of as a part of the general area of context-awareness. In our scenarios the contexts are nothing but the situations and activities of users. These contexts are typically sensed by a variety of sensors residing in the users' environment. In case of the cell phone scenario many of these sensors would reside on the system (i.e. the cell phone) itself. The section on scenarios (and implementation issues) lists a variety of sensors that may be used for such architectures. The sensors provide input in the form of information regarding user activities and situations. This information is employed by the Context-Aware Component (CAC) as training data for building models to classify new situations and activities based on interruption awareness protocols. The core of the CAC is a machine learning algorithm (in our case, a J48 decision tree algorithm) that builds the models based on the training data.

Once the model is built it can be used by other application components in the system to provide interruption aware services. For example, after recognizing a situation,

the results obtained from the CAC learning model could be used in changing the notification modes or providing appropriate responses to incoming visitors. The interaction between the CAC and the rest of the application may occur via a host of APIs. Knowledge and databases are also employed to store a variety of information, ranging from sensor data to learned concepts and situations. One point to be noted here is that we envision the context (or interruption) aware functionality to be designed such that it works in parallel with other application functions and services. The figure shows two-sided arrows connecting the sensors and the I/O with both the CAC and rest of the application. An interruption aware cell phone would work as a normal cell phone but would also provide additional interruption handling functionality by tapping into the incoming data in parallel. Figure 38 shows a more detailed look at the architecture for simulating the cell phone and virtual secretary scenarios.

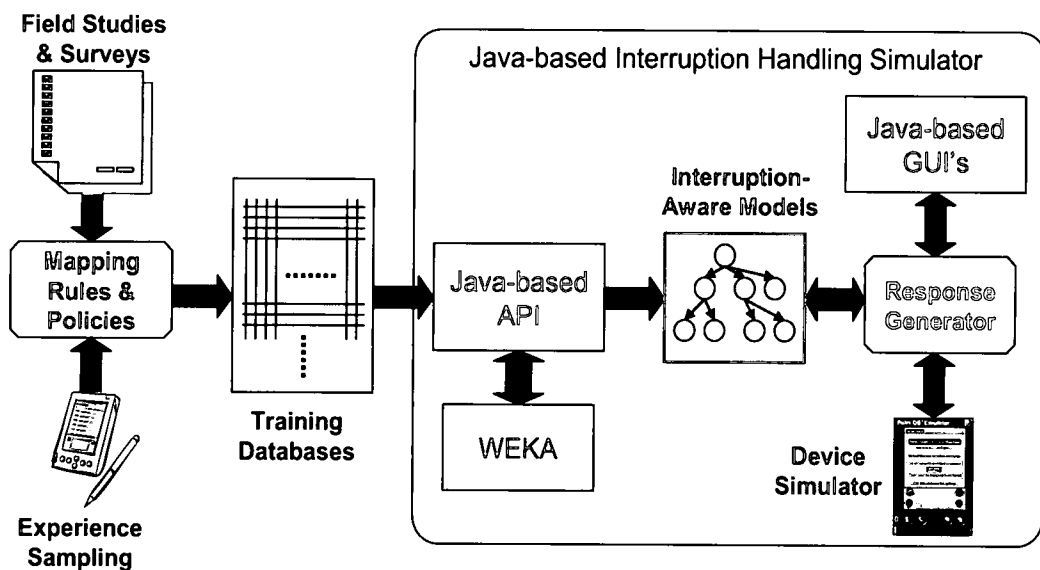


Figure 38. Architecture of the Java-based Interruption Handling Simulator

Platform: Windows XP, 2GB RAM, 3.0GHz Intel Processor
Programming Language: Java SDK version 1.4.2_08
Algorithm: J48 Decision Tree Learner
Algorithm Source: WEKA Machine Learning Suite version 3.4
Training Data: ARFF formatted text files
Interruption Aware Model: Java based
User Interfaces: Java based GUI's
Experience Sampling Tool: Palm Tungsten E
Experience Sampling Tool Platform: Palm OS 5.0
Experience Sampling Tool Programming Language: Java J2ME Wireless Toolkit
Mapping Rules and Policies: Manual (statistics and designer preferences)

Table 19. Summary of Tools Employed for Developing the Simulation Architecture

Table 19 shows a summary of the tools employed for developing this architecture. The simulator was developed entirely in Java and as the architecture in figure 44 shows it also interfaces with external components such as class libraries, device simulators, and storage files. The core of the simulator is the interruption-aware model built by a J48 decision tree learning algorithm. The algorithm is a part of the WEKA [Witten et al., 2005] software development suite. The model uses training data coming from user surveys, field studies and experience sampling to classify new situations and respond in an interruption aware manner. The training data is built by converting the information obtained from the user studies into rules and policies. It is currently stored as text files but it is quite possible that actual implementations would employ databases for storage purposes. Also, the process of obtaining information from user studies and creating the datasets is currently manual in nature but as we will see in the section on future work, this process can also be automated.

The Java based APIs were designed to interface between the training datasets and the WEKA based J48 classifier algorithm. The resulting model was stored locally for

future use. As the figure suggests, the model interacts with the simulator front end via a response generator. The response generator is essentially a mapping structure that generates output responses according to the results produced by the model. The response generated also interfaces with a host of GUIs that simulate actual device (or system) user interfaces capable of input and output. Thus, whenever a new situation occurs, the model classifies it and produces a result. The response generator uses this result to invoke specific outputs in the GUIs. For example, if a user goes into a theater and receives a call the model may recognize it and produce a result. This result is used by the response generator to change the interruption modality from ringer to vibrate. In the next section we look at how the decision tree learning is used for interruption handling in the simulator.

7.2. Enabling Technologies

The selection process of simulation enabling technologies is slightly different from that of the selection process of technologies for actual system implementation. For modeling and simulation, designers are generally looking to build a software based design that will provide an accurate representation of the proposed design. They are also looking to address issues such as performance, functionality, services offered, interaction, time, etc. and the tools required to provide information about these issues may vary. In the following paragraphs we will look at some of the tools and techniques that were employed for creating the interruption aware models and designing the simulator. In the section on implementation issues however we will look at the selection choices and

issues to be considered when choosing enabling technologies for the actual implementation of interruption aware systems.

7.2.1. Java SDK and J2ME

Simulator design can be performed either by employing existing simulator environments and widgets or designing the entire simulator from the beginning. The problem with using existing simulator tools is that they are not application and scenario specific. Furthermore, it is very difficult to integrate their design components with third party tools. In designing the simulator for the cell phone scenario there were many existing choices for simulator tools [Nokia] [Palm] that provided a comprehensive set of functionality and services related to wireless connectivity. Apart from having a resourceful programming environment (the Java J2ME), they also had good phone simulators with considerable functionality. Unfortunately, they did not provide the interface choices that were required for the cell phone scenario and more importantly, did not have APIs to interface with third party tools such as WEKA which was designed using the Java SDK. For the virtual secretary scenario, no existing environments or simulators were found.

In order to develop a single simulator that would provide modeling and simulation ability for both the cell phone and virtual secretary scenarios, the Java Standard Development Kit programming environment was chosen. Java SDK was chosen due to its highly effective object-oriented design style which allows complex designs and interfaces to be written in a simple and uncomplicated manner. Java is also portable and this allowed for the simulators to be subsequently tested on both Windows and UNIX

platforms for performance related issues. The Swing APIs for Java allowed effective designs of the phone simulators and screens for viewing the simulation responses for both the cell phone and virtual secretary scenarios. The only disadvantage with using Java as the programming environment was that the code was executed on the JVM instead of directly on the system hardware which forced the heap sizes to be increased close to 1GB for creating the decision tree models for large training datasets.

The J2ME platform is a collection of APIs designed specifically for resource constrained mobile and wireless devices. J2ME supports the CLDC (Connected Limited Device Configuration) and the MIDP (Mobile Information Device Profile) protocols which were developed specifically for resource constrained devices. J2ME consists of a smaller set of classes and libraries designed specifically for connectivity and graphics programming on cell phones and PDAs. For the purposes of this research, J2ME was confined to testing and experience sampling instead of designing the entire simulator for reasons discussed above. While none of the simulator interfaces or screens were designed using J2ME, it was nonetheless employed for programming Palm PDAs for conducting experience sampling studies. It allowed quick and easy creation of forms and surveys and their installation on the PDAs. The data stored from the experience sampling studies was also easily retrievable enabling it to be analyzed and converted into training datasets.

7.2.2. WEKA

The WEKA suite [Witten et al., 2005] is a collection of state-of-the-art machine learning tools and algorithms integrated into a GUI based experimentation environment. It allows the construction and testing of models that are built by these algorithms. It also

allows designers to perform a comprehensive set of data mining tasks such as preparing the input datasets, model and algorithm evaluation, and so on. An added advantage of using WEKA is that it allows effective visualization of the experiments and their results in both graphical and numeric formats.

WEKA is written entirely in Java and is capable of running on UNIX, Windows, and Mac operating systems. This was advantageous to the simulator design because it was also written in Java making it possible to use classes from WEKA directly in the simulator code with minimal APIs. For example, the decision tree classifier J48 was used in the simulator code by calling its methods and classes for performing classification and evaluation tasks. In summary, WEKA allows for the following types of activities to be performed via a GUI based experimenter tool or permits the use of the corresponding classes in independent Java programs [Witten et al., 2005].

- Classification and regression (ID3, RandomForest, NaiveBayes, etc.)
- Clustering (Simple K-Means, Density Based, etc.)
- Association rules (Apriori, Predictive Apriori, etc.)
- Attribute evaluation (Information Gain, Chi-Squared, Wrapper based, etc.)
- Search (Best First, Greedy, Stepwise, Ranker, etc.)
- Filtering (Attribute Selection, Stratified Remove, Resample, etc.)
- Data visualization (trees, probability tables, graphs, etc.)

WEKA provides tools and algorithms for developing both supervised and unsupervised learning models. The training data used for supervised learning algorithms needs to be in the ARFF format for it to be recognized by the classifier algorithms.

WEKA algorithms are capable of processing both categorical and numeric datasets. A sample dataset in the ARFF format is shown below.

```
@relation vsec

@attribute conversation {none, self, other}
@attribute pressure {occupied, unoccupied}
@attribute schedule {yes, no}
@attribute time {workhours, officehours, lunchhours}
@attribute phone {yes, no}
@attribute visitor {student, colleague, vip, other}
@attribute purpose {appointment, dropin, emergency}
@attribute allowvisit {yes, no, wait}
@data

none,occupied,no,workhours,no,student,appointment,wait
self,occupied,no,workhours,no,student,dropin,wait
other,occupied,no,workhours,no,student,dropin,wait
other,occupied,no,workhours,no,colleague,dropin,no
other,occupied,no,workhours,no,student,dropin,wait
none,occupied,no,workhours,no,student,dropin,wait
other,occupied,no,workhours,no,colleague,dropin,yes
none,occupied,no,workhours,no,student,dropin,no
none,occupied,no,workhours,no,student,dropin,no
none,occupied,no,workhours,no,student,appointment,wait
none,occupied,no,workhours,no,colleague,dropin,yes
none,occupied,no,workhours,no,student,dropin,no
other,unoccupied,no,lunchhours,no,student,dropin,wait
other,occupied,no,workhours,no,student,dropin,no
other,occupied,no,workhours,no,student,dropin,no
other,occupied,no,workhours,no,student,dropin,no
none,occupied,no,workhours,no,colleague,appointment,yes
```

Incorporating WEKA classifiers and other tools is as easy as including the appropriate class libraries in the program code as follows:

```
import weka.classifiers.*;
import weka.classifiers.trees.j48.*;
import weka.classifiers.trees.*;
import weka.core.*;
```

Once this is done, the simulator can interface with the WEKA tools by calling the methods directly and providing them with the appropriate variables. The algorithms themselves can be modified and refined for application specific requirements by changing a host of existing flags and options such as random seed, number of folds, and pruning parameters.

Evaluation using WEKA allows designers to perform a variety of activities. Learned models can be evaluated for their accuracy in classifying instances. This accuracy is returned in the form of error rate, false positives and confusion matrices. Attributes can also be evaluated to determine their worth in building the learning model. Attribute evaluation usually takes place by searching for and ranking attributes based on a variety of available techniques such as information gain.

In the next section we will look at the decision tree algorithm that was used as the primary learning algorithm for developing the interruption aware models for the cell phone and virtual secretary scenarios.

7.2.3. Modeling using J48 Decision Trees

In order to develop the interruption aware learning models for the two scenarios the WEKA J48 decision tree algorithm was employed. J48 is a WEKA version of the highly popular and robust C4.5 learner [Quinlan, 1986] [Quinlan, 1993]. C4.5 itself is an extension of the ID3 learning algorithm and accounts for missing values, continuous attributes, pruning, rule derivation, etc.

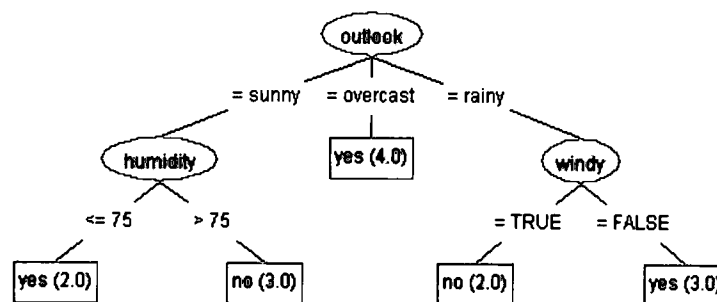


Figure 39. A Sample J48 Decision Tree

Figure 39 shows a sample decision tree created by employing the J48 learner on a training dataset to classify whether it is a good day to play tennis. As the figure shows, nodes represent attribute choices and the paths represent choice of attribute values that were selected by the algorithm in coming to a decision. The training dataset for this decision tree is shown in Table 20.

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
=====				
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

Table 20. Training Dataset for Decision Tree in Figure 45.

We notice that J48 selects *outlook* as the starting (or root) node for constructing the decision tree. This selection is based on the entropy of each attribute. In other words, an attribute that is the most informative towards building a tree is used as the root node. Information gain is employed to measure how informative an attribute is. It describes the expected reduction in entropy by selecting a particular attribute as the node.

$$E(I) = -\frac{P}{p+n} \log_2 \frac{P}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

$$\text{Gain}(A, I) = E(I) - \sum_{\text{outcome } j} \frac{P_j + n_j}{p+n} E(I_j)$$

$E(I)$ and $Gain(A,I)$ represent the entropy and expected information gain respectively. J48 uses this principle at every node during the decision tree construction until all attributes have been covered and no attribute is repeated in the same path.

There are various classes in WEKA that can be used for building decision trees. The following illustrates one way of instantiating a decision tree classifier using the J48 algorithm in WEKA.

```
Classifier c = new Classifier();  
c = Classifier.forName("weka.classifiers.trees.J48",options);  
c.buildClassifier(train_file);  
:  
:  
:  
response_class = c.classifyInstance(instance_file);
```

We notice here that we first need to instantiate a classifier and indicate what type of classifier algorithm we want to use to build the decision tree model. Once this is done we need to provide it a training dataset which is stored in an 'arff' formatted text file. The decision tree model that is built in this manner can be stored for future classification purposes. The classification of new instances takes place as shown in the last line, i.e. by calling the created model, and providing it with a new and previously unseen set of instances which are also stored as text files.

J48 also allows a host of options for refining and pruning the decision tree models for increased accuracy. These options allow designers to set the confidence threshold for pruning, minimum number of instances per leaf, number of folds, and so on.

A variety of evaluation options can be used along with the J48 to analyze the accuracy of the models that are built. Figure 40 shows a GUI output of the J48 model construction and evaluation for the example discussed above.

```

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      9           64.2857 %
Incorrectly Classified Instances    5           35.7143 %
Kappa statistic                    0.186
Mean absolute error                 0.2857
Root mean squared error             0.4818
Relative absolute error             60 %
Root relative squared error         97.6586 %
Total Number of Instances          14

=== Detailed Accuracy By Class ===

TP Rate    FP Rate    Precision    Recall    F-Measure    Class
  0.778      0.6       0.7         0.778     0.737       yes
  0.4        0.222     0.5         0.4       0.444       no

=== Confusion Matrix ===

a b  <-- classified as
7 2 | a = yes
3 2 | b = no

```

Figure 40. GUI Output of Decision Tree Model Results

As the figure shows the time taken to build model, the number of correctly and incorrectly classified instances, the error rate, the true and false positives, and the confusion matrix are displayed for analysis. The confusion matrix is simply a table showing correctly and incorrectly classified instances. In this case, we see that two instances were classified incorrectly as 'b' instead of 'a', and three instances were classified incorrectly as 'a' instead of 'b'. In the section on training data analysis we will see how the error rate varies with the size of the dataset.

Other evaluation procedures that can be performed are attribute selection and filtering. These procedures rank attributes based on their usefulness in creation of the model. Reduction in attributes could greatly reduce the size of the training datasets and maintain or even increase the accuracy of the generated models. We will discuss this in further detail in the section on the evaluation of attributes.

Decision tree pruning can also be performed to reduce the size of the decision tree model and increase accuracy by reducing errors caused due to overfitting. By default the J48 algorithm uses the C4.5 pruning technique and as we will notice in the evaluation section, the accuracy was high enough that additional pruning was not required. In the next section we look at other technologies such as the Palm OS and Java for mobile devices which were used for testing the experience sampling methods for data collection.

7.2.4. Palm OS and IBM WEME

Experience sampling studies were conducted using the Palm Tungsten E handheld PDA. This device was chosen due to its processing power, onboard memory and flexible interaction with the underlying operating system and its components. The Tungsten E ran on the Palm OS version 5.0 with the IBM WebSphere Everyplace Micro Environment (WEME) installed on it. WEME is a certified Java product developed under agreement between IBM and Sun Microsystems. WEME enables a wide variety of applications and services developed in Java to run on the Palm OS and have access to the underlying system functionality and interfaces. The advantage of WEME is that it is supported not only on the Palm OS, but also on other competing systems such as Windows, Linux,

PocketPC and Symbian. This provides a greater choice while selecting the handheld devices.

The core of WEME is the J9 virtual machine, an IBM implementation of the original Java VM which is essential to run any application written in Java. The J9VM was designed specifically for resource constrained devices such as mobile phones and handhelds. The J9VM along with a standard set of the Java Class Libraries offer support for CLDC and MIDP technologies which enable designers to write a variety of applications for resource constrained devices. Such applications are capable of wireless communication, resource access and sharing, graphics based services, and document management.

Storage on the Palm OS based devices is designed as a collection of records, and accessing or writing to these records involves working on their attributes, IDs and locations. This type of storage mechanism proves effective for context-aware systems because sensor information is generally categorized in nature and can be easily stored and retrieved from such devices without substantial modifications.

While Palm OS based programming and devices provide a variety of options their main disadvantage comes in the limitation of the Java class libraries. Due to this, standard Java simulator code such as the WEKA model generation or classifier testing could not be ported onto the handheld. This was the main reason why the actual device programming was limited to experience sampling studies. However, specialized interfaces called conduits are available that provide low level interaction between the device and other systems such as PCs.

7.3. Experimentation

In this section we will look at the working of the simulator with the help of experimental runs, screens, and testing of sample situations. The following section looks at the different screens and the sequence in which the simulator is able to test the cell phone and virtual secretary scenarios.

Figure 41 shows the first screen of the simulator. This screen contains the following options:

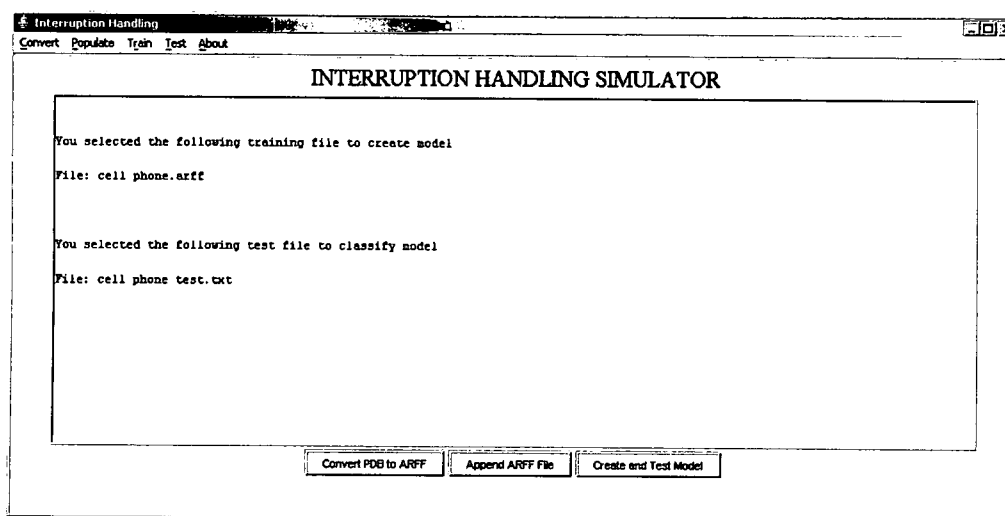


Figure 41. Screen 1 Displaying Basic Options for Data Creation, Conversion and Selection

- Convert data obtained from experience sampling tools to appropriate formats which can be then used in the creation of decision tree models for the cell phone and virtual secretary scenarios.
- Create training datasets for developing decision tree models by entering the attributes and their values.
- Create storage files for test datasets for testing various instances and situations.

- Select both training and test dataset files for creating and testing the decision tree models of the cell phone and virtual secretary scenarios.
- Activate user interfaces for creation and testing of decision tree models.

Once the training datasets and test datasets have been created they can be selected using the options in the first screen. After this, the “Create and Test Model” button is clicked which activates the second screen (Figure 42) which enables the user to select between one of three algorithms. Currently only one algorithm is implemented, namely the J48 decision tree algorithm.

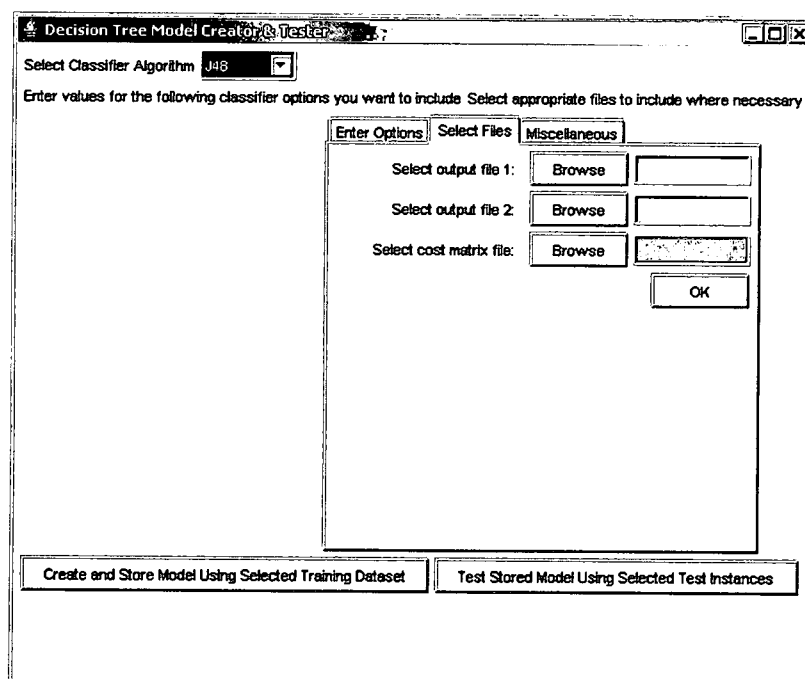


Figure 42. Screen 2 Displaying Options to Create, Select and Test the Decision Tree Models of the Two Scenarios

Once the J48 algorithm is selected another GUI is activated which has options for running the J48 algorithm on the selected training datasets to create the decision tree models. Options for selecting where the created decision tree models can be stored are

also available. There are also options for activating the simulator screen and testing the decision tree models on previously selected test instances.

Figure 43 shows the simulator screen that is activated when the decision tree models for the training data (cell phone or virtual secretary) are being created

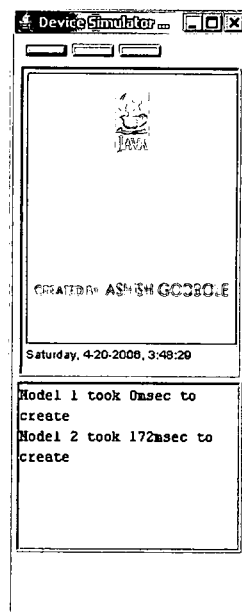


Figure 43. Screen Showing Default Simulator Screen for Displaying Results of the Interruption Handling Process

As the figure shows, the results of classifying new instances (i.e. handling interruptions on a case by case basis) are displayed on the simulator screen. Additional text information is provided about model creation, testing and responses. Figure 44 shows a table that displays test instances and their corresponding response class as determined by the interruption handling decision tree model. This table corroborates with the result obtained in figure 43.

The decision tree model for the virtual secretary scenario uses 8993 instances for training from a possible set of 409600 instances. This reduction was achieved by manual and rule based selection criteria that depend on user information elicitation and designer preferences.

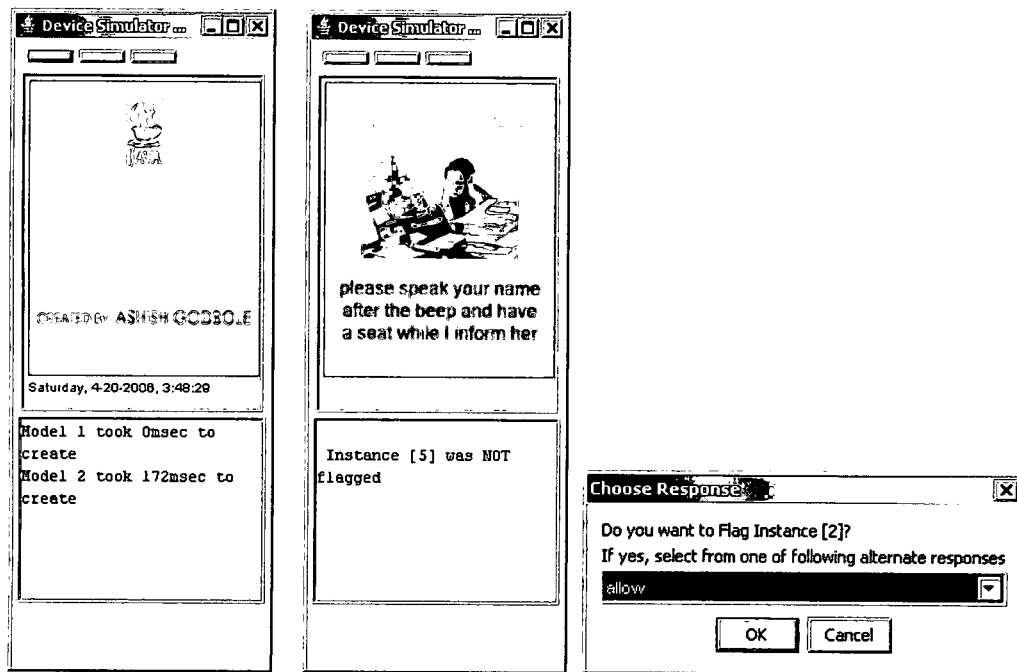


Figure 45. Simulator Screen Displaying Change in Graphics (and Audio) for Each Newly Tested Instance for the Virtual Secretary Scenario

Figure 45 and Figure 46 illustrates the different screens after the model has been created and each of the instances has been tested. Notice that the first and fifth (last) instances in the sample test set are the same. When the first instance occurred, the user changed its response preference from 'checklater' to 'allow'. When this instance occurred again (the fifth in the set), we see in the table that the system uses the user's preferred response instead of the original one. In this manner, post-design information elicitation can be incorporated in the system.

in table 21 above. In the next chapter we will look at some testing and evaluation studies performed to ascertain factors such as accuracy, size, attribute relevancy, and so on. In this chapter we will also validate the simulators by having end users test it and compare their preferences with the results obtained from the simulator.

CHAPTER 8

EVALUATION AND VALIDATION OF SIMULATOR

In this chapter we will look at the next stage in the design process of context and interruption aware systems, namely evaluation and validation. While not the last stage of the system design process, this is the last stage in the modeling and simulation of the design scenarios. It is an essential stage because it allows designers to determine whether their approaches, architectures and techniques were effective before they can start implementing the system. Evaluation and validation of models, simulations or prototypes serves the following purposes.

- Functionality of the proposed system in terms of services provided, user satisfaction and goals achieved can be determined.
- Performance of the proposed system in terms of various evaluation parameters such as speed, responsiveness, errors rate, etc. can be calculated.
- Effectiveness of the design approach (in this case the user information driven decision tree model) can be determined.

This process can be divided into two categories; testing and evaluation of the system and end-user validation. We look at each in detail below.

8.1. Evaluation of the Simulator

Testing and evaluation of the generated system models or simulations is important because the results generated are used in the further development of the system, i.e. in the implementation stages for selection of enabling technologies, refining design functionality, policies and rules, and so on. This section discusses the evaluation of the simulators based on a variety of performance parameters such as the size of training data and models, speed and time for creation of models, testing times for new instances, and attribute effectiveness. The section solely focuses on the virtual secretary scenario evaluations because the cell phone scenario could not be evaluated due to memory constraints owing to extremely large training datasets. However, the process of evaluations is the same and can be applied to the cell phone scenario as well.

8.1.1. Training Data Analysis

In this section we look at the training datasets that were generated from the set of rules and policies obtained from user surveys, field studies and experience sampling. Information obtained from users is highly unorganized and we employ statistical techniques and direct observation to glean rules and policies out from this information. In order to create effective training datasets these rules and policies must be accurate and generic enough to cover a large number of cases, instances and situations. Otherwise the datasets that are generated will either be very small and inconsequential, or will be very large and cost prohibitive towards building the learning models. Thus the selection of optimal training datasets in terms of size and performance is essential. Below we compare how the size of the training datasets for the virtual secretary scenario affects the

performance of the learning model in terms of the time taken to build the model, the model error rate and other issues.

Experiment Type
1% Training Data
5% Training Data
10% Training Data
20% Training Data
30% Training Data
40% Training Data
50% Training Data
60% Training Data
66% Training Data
70% Training Data
80% Training Data
90% Training Data
95% Training Data
10-fold Stratified Cross Validation

Table 22. Varying the Training Dataset Size for Comparison of Model Performance

Table 22 shows fourteen experiments with three runs each that were carried out in order to measure the performance of the generated decision tree model for the virtual secretary scenario. For each of these fourteen experiments, the instances that were selected for the training dataset were chosen both manually as well as in a completely random manner. Thus we have two kinds of results, one showing the performance when data is chosen manually and another showing the performance when data is chosen at random. Figure 47 shows how the two training datasets (manual and random) were varied for the fourteen experiments.

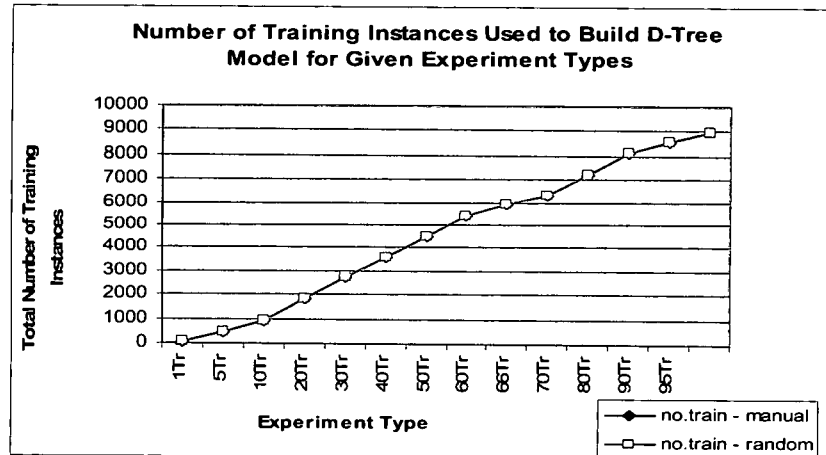


Figure 47. Variation of Dataset Size for Both Manual and Randomly Sampled Experiments

From Table 19 we see that the final training dataset for the virtual secretary was reduced drastically from 409600 instances to only 8993 instances. This was possible due to many physically impossible cases and cases that were not logical, such as when 'chairpressure' is 'yes', and 'presence' is 'no', or when 'currentvisitor' is 'none', 'soundtype' is 'speech', and 'onphone' is 'no'. There are five response classes in the virtual secretary scenario and their distribution in the training dataset is given in table 23.

Total number of possible instances for virtual secretary scenario: 409600
Number of instances in full training dataset: 8993
Total number attributes: 12
Attribute values: non-binary
Number of response class values: 5
Distribution of class values in full training dataset:
allow: 5073 (57%)
checklater: 1988 (22%)
wait: 714 (8%)
takemssg: 526 (5%)
donothing: 692 (8%)

Table 23. Properties of the Training Datasets for the Virtual Secretary Scenario

The graphs below show the results from the fourteen experiments for various sizes of training data as well as various kinds of sampling, i.e. manual, random, and cross-validation.

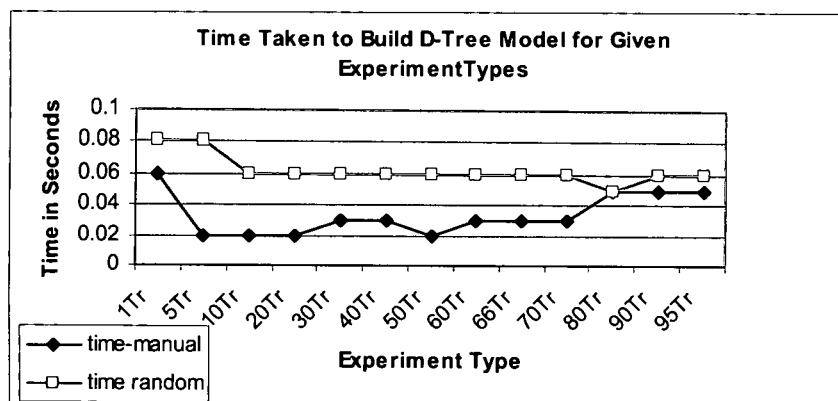


Figure 48. Time Taken to Build Decision Tree Model for Manually Sampled and Randomly Sampled Training Datasets

Figure 48 shows the comparison of the time taken (in seconds) to build the decision tree model for both the manually sampled as well as the randomly sampled training datasets for each of the fourteen experiments. The time taken to build the model based on manual sampling seems to be lower because the samples were non-random, i.e. they were chosen in clusters such that many of the same response classes were grouped together in the training dataset. The conclusion is that it generally takes longer time to build a model from a completely random dataset than a manually sampled dataset. We also notice that as the size of the dataset increases, the time difference decreases. This may be due to the fact that as the size increases the two datasets (manually sampled, and randomly sampled) start to resemble each other.

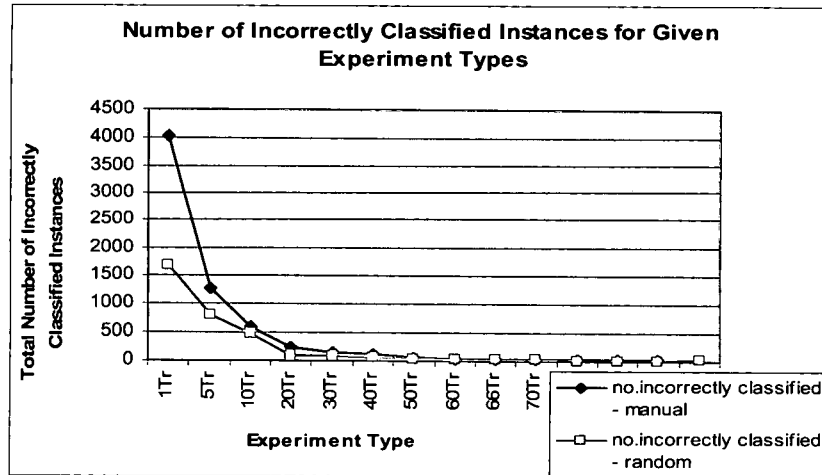


Figure 49. Classification Accuracy for Manual and Randomly Sampled Datasets

Figure 49 shows the total number of incorrectly classified instances for different dataset sizes for both manually and randomly sampled training datasets. It can be noticed here that the number of incorrectly classified instances decreases drastically from 1 percent dataset size to approximately 40 percent after which the decrease is steady. From this we notice that manually sampled training datasets have more incorrectly classified instances than randomly sampled training datasets. This conclusion seems appropriate because the manually sampled training datasets are created in a clustered fashion such that many of the response classes are grouped together during the instance selection process. Thus it is possible that the class distribution changes and is different from the original class distribution. In other words, when the decision tree model is built, some classes may be preferred over other classes leading to a the higher number of incorrectly classified instances. This is not true in the case of randomly sampled training datasets, because during their creation care was taken to keep the class distribution constant. We

also notice that as the training dataset size increases, this effect is diminished because the two datasets (manual and random) start looking similar to each other. In Figure 50 we notice that as the size of the training dataset increases the accuracy of the model, i.e. the percentage of correctly classified instances increases drastically, and then levels off. This effect can be noticed further in Figure 51 which shows the mean absolute error rate for the two datasets for all fourteen experiments.

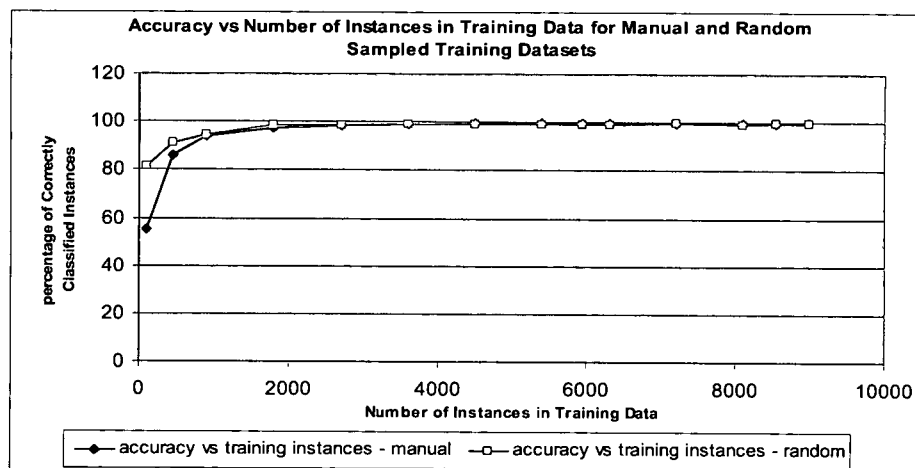


Figure 50. Accuracy of Model for Varying Training Dataset Sizes

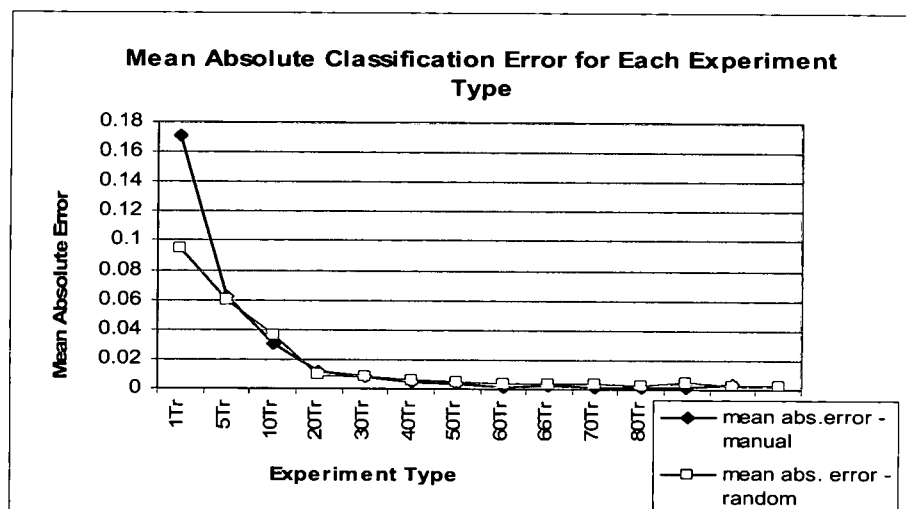


Figure 51. Mean Absolute Error Rate for Virtual Secretary Scenario

From these figures we notice that the training dataset provides lesser and lesser information as its size increases. We notice that at about 40 percent of the original dataset, an optimal performance in terms of accuracy is achieved. Table 24 illustrates this in numeric form.

Number of incorrectly classified instances for 1% training dataset size	
Manual Sampling:	4036
Random Sampling:	1670
Number of incorrectly classified instances for 30% training dataset size	
Manual Sampling:	132
Random Sampling:	99
Number of incorrectly classified instances for 40% training dataset size	
Manual Sampling:	104
Random Sampling:	47
Number of incorrectly classified instances for 50% training dataset size	
Manual Sampling:	64
Random Sampling:	35
Number of incorrectly classified instances for 90% training dataset size	
Manual Sampling:	26
Random Sampling:	10

Table 24. Number of Incorrectly Classified Instances for Various Dataset Sizes

Here we see the decrease in incorrectly classified instances for some relevant dataset sizes for both manual and randomly sampled training datasets. In other words, an optimal training dataset for the virtual secretary scenario is a completely randomly sampled dataset with a size that is about 40% of the original dataset. In real time data collection schemes such datasets are achievable and the randomness of the datasets can also be maintained by careful data collection. We notice that a reduction in dataset size

from 406900 to 8993 (about 40%) can be achieved by carefully weeding out the impossible cases and evaluating the performance of various datasets created via random and manual sampling techniques. Similar studies can be done on the cell phone scenario. In the next section we will look at another factor that enhances the optimization of a training dataset for the creation of the learning model. That is the evaluation of attributes for proper selection of sensors and attributes.

8.1.2. Evaluation and Selection of Attributes

Attribute evaluation is a very important task that needs to be performed early in the design process. This enables the designs to be more accurate and allows for the selection of enabling technologies and infrastructure that are appropriate to the scenarios and domains. For the purposes of this research, attribute evaluation after the modeling and simulation phase allows designers to be able to select the right kind of sensors and attribute value pairs for further design and implementation stages. In the following paragraphs we will look at some attribute evaluation techniques and experiments that were carried out for the virtual secretary scenario.

Attribute evaluation and selection can be performed manually or by employing specific techniques. Manual methods are exhaustive and as the data or attribute set increases they become very tedious and time consuming. They are also very inaccurate in nature. Attribute evaluation techniques on the other hand are much less time consuming, and precisely evaluate and select attributes based on a variety of mathematical techniques. They can be broadly divided into two categories; individual attribute evaluators, and attribute subset evaluators. The techniques in the first category evaluate attributes

individually based on their predictive ability, while the techniques in the second category evaluate various subsets of attributes to ascertain the best performing subset. WEKA has a variety of algorithms from both categories to assist in the attribute evaluation and selection procedure. Table 25 briefly describes some of these algorithms and their operating principles [Witten et al., 2005].

Algorithm Name	Description
CfsSubsetEval	Considers predictive value of each attribute in a subset, along with the degree of redundancy between intra-subset attributes.
ClassifierSubsetEval	Employs user selected classifier to evaluate a subset of attributes. Performs classification on a hold out set that is separate from the training dataset.
WrapperSubsetEval	Employs a user selected classifier to evaluate a subset of attributes, but performs cross-validation instead of using a holdout set.
InfoGainAttributeEval	Evaluates individual attributes in the entire dataset based on the principle of Information Gain and Entropy
ReliefFAttributeEval	Evaluates individual attributes in the entire dataset based on comparison between nearby instances.

Table 25. Attribute and Attribute Subset Evaluation Algorithms in WEKA

Once the set of attributes has been evaluated based on predictive ability or classification accuracy, analysts can also choose to search for the best attributes and rank them based on which attribute provides the highest amount of information towards generating the learning model. To assist in this WEKA provides a variety of attribute ranking techniques. Table 26 briefly describes some of these algorithms and their operating principles [Witten et al., 2005]. As the table shows there are a variety of ranking techniques. Some of these techniques work based on greedy searching schemes, while others are random, while still others are exhaustive in nature. Some techniques employ multiple searches and perform inter-search comparison to select the best search based on t-tests.

Algorithm Name	Description
BestFirst	Greedy hill-climbing with backtracking.
ExhaustiveSearch	Search exhaustively through the entire dataset.
GeneticSearch	Search using a simple genetic algorithm.
RandomSearch	Search randomly through the entire dataset
RankSearch	Sort attributes and rank promising subsets using an attribute subset evaluator.
Ranker	Rank individual attributes (not subsets) according to their evaluation.

Table 26. Attribute Search and Ranking Algorithms in WEKA

From the above table we notice that all the listed techniques are search techniques except the last one which is a ranking technique. In WEKA, ranking techniques are employed in combination with the individual attribute evaluation techniques such as 'InfoGainAttributeEval', or 'ReliefFAttributeEval'. The search techniques are employed in combination with the attribute subset evaluators in WEKA such as the 'CfsSubsetEval', or the 'WrapperSubsetEval'. Below we will see results of the attribute evaluation and selection procedures carried out to determine which attributes provide the least amount of information towards developing the learning models, and which are the most redundant.

As the Table 27 illustrates, the two attribute subset evaluation techniques employed are 'CfsSubsetEval' and 'WrapperSubsetEval' respectively. The first technique considers attribute subsets as relevant only if the attributes in the subsets are minimally correlated to each other and maximally correlated with the response class. The second technique employs the J48 decision tree learning algorithm to classify the attribute subsets that are generated by the attribute evaluator. This classification is performed every time attributes are added and the change in performance on every attribute addition or removal determines how useful that attribute was to the determination of the class.

CfsSubsetEval RankSearch	WrapperSubsetEval RankSearch
9 purpose 7 calendar 8 newvisitor 12 awaytime 5 presence 1 onphone 11 busytime 10 currentvisitor 6 busyattribute 2 motion 3 chairpressure 4 soundtype	9 purpose 7 calendar 8 newvisitor 12 awaytime 5 presence 1 onphone 11 busytime 10 currentvisitor 6 busyattribute 2 motion 3 chairpressure 4 soundtype

Table 27. Ranked Results of Two Attribute Subset Evaluation and Selection Techniques

The 'RankSearch' algorithm is employed in both cases to rank the attribute set from the most informative attribute to the least informative attribute. Results show that the least informative attributes for both methods are 'motion', 'chairpressure' and 'soundtype', while the most informative attributes are 'purpose', 'calendar' and 'newvisitor'. Based on this, the best performing subset is obtained which is the set of first nine values in table 27 for each of the methods. The last three are subsequently discarded.

InfoGainAttributeEval Ranker	ReliefFAttributeEval Ranker
8 newvisitor 9 purpose 1 onphone 11 busytime 10 currentvisitor 12 awaytime 5 presence 6 busyattribute 7 calendar 2 motion 3 chairpressure 4 soundtype	8 newvisitor 9 purpose 1 onphone 11 busytime 6 busyattribute 10 currentvisitor 12 awaytime 5 presence 7 calendar 2 motion 3 chairpressure 4 soundtype

Table 28. Ranked Results of Two Individual Attribute Evaluation and Ranking Techniques

Table 28 shows results of the two individual attribute evaluation techniques employed in combination with the 'Ranker' algorithm. The first technique 'InfoGainAttributeEval' employs information gain and entropy principles for each individual attribute to ascertain how useful it is towards building the learning model. The 'ReliefFAttributeEval' method is an instance based evaluation technique that looks at the nearby instances to predict the usefulness of an attribute. In both the cases, the 'Ranker' algorithm is employed to rank the individual attributes based on how informative they are to creating the model.

The results show that in this case too, the most redundant and least informative attributes are 'motion', 'chairpressure' and 'soundtype'. The ranking suggests that the best attribute 'newvisitor' is different from the one obtained using attribute subset evaluator methods (i.e. 'purpose') but overall however, the rankings are fairly consistent across the methods. The slight differences in rankings could also be due to the fact that the attribute subset evaluator methods are correlation based while the individual attribute methods are based on how informative the attribute is to the learning model.

After removing these attributes from the virtual secretary dataset, a 10 fold cross validation experiment was conducted to compare the performance of the original dataset and the filtered dataset. It was noticed (refer to Table 29) that there were only 41 instances which were incorrectly classified compared to 43 in the original dataset, a difference in classification percentage of 0.022 which is highly insignificant given the size of the data. On the other hand, if we filter out an informative attribute such as 'newvisitor' along with the other three, the difference in classification percentage jumps to 33.170 percentage points.

Original Training Dataset (all 12 attributes)
Number of incorrectly classified instances: 43
Classification percentage: 99.5219%
Filtered Training Dataset (<i>minus motion, chairpressure, soundtype</i>)
Number of incorrectly classified instances: 41
Classification percentage: 99.5441%
Percentage Difference: 0.022
Filtered Training Dataset (<i>minus motion, chairpressure, soundtype, busyattribute</i>)
Number of incorrectly classified instances: 581
Classification percentage: 93.5394%
Percentage Difference: 5.982
Filtered Training Dataset (<i>minus motion, chairpressure, soundtype, newvisitor</i>)
Number of incorrectly classified instances: 3026
Classification percentage: 66.3516%
Percentage Difference: 33.170

Table 29. Comparison of Classification Percentage for Original and Reduced Training Datasets for the Virtual Secretary Scenario

This suggests that the three attributes that were discarded were inherently uninformative and redundant to the dataset. Similar techniques can be employed for the cell phone scenario which has a much larger training dataset in order to reduce the dimensionality and increase the accuracy of the learning model.

8.2. User Validation

In the previous sections, we discussed issues related to system performance, functionality and accuracy related to how well it achieves the proposed goals. This evaluation is not sufficient because even though the system works correctly and performs well, it has not been tested by the end-users yet. The main reason for evaluation using end-users is to ascertain if the system performs well in a real environment, and whether it is satisfactory to its intended users. Unfortunately models and simulators provide a very

low scope for end-users to test and validate the design. However, in the absence of actual prototypes, they remain the best means of end-user validation.

The validation process involves having end-users sit through and perform simulation runs, cover main use cases and scenarios, and determine whether the simulator results are in accordance with their preferences. In this section we will look at the process and results of end-user validation of the cell phone and virtual secretary simulations to determine if the interruption handling method used was effective from a user perspective. The sequence of steps performed for user validation is as follows:

- Seat the user at the simulation desk and run the simulator to the point where it is ready to accept input and display results.
- Give the user a survey that lists the different use cases and scenarios.
- Explain to the user the scope and limitations of the use cases and scenarios.
- Inform the user on what she can expect from the simulator in terms of interruption results and choices available.
- Ask the user to fill out a survey item on how she is currently being interrupted in the given use cases and scenarios.
- Ask the user to fill out survey item on how she would prefer to be interrupted for each of the use cases and scenarios.
- Run each of the use cases and scenarios by feeding appropriate instances as input to the simulator.
- Observe the simulator results and ask the user whether she is satisfied with them, or whether she would have preferred another response.

The following factors determine the effectiveness of the interruption handling method.

- Determine the similarity between the user's current settings (default settings in case of cell phones and default protocols in case of office secretaries), her preferred choices (settings she believes would reduce interruptions) and the simulator results obtained by using the interruption handling method.
- Ideally, the user's preferred choices, and results obtained from the simulator must be the same or as close to each other as possible. This would indicate that the simulator works satisfactorily in accordance with the user's preferences. A large difference would indicate that the simulator produces results that are considerably different from the user's preferences.
- The user's original interruption responses may or may not be different from her preferred responses to interruptions. A small difference would mean that she is happy with the way her system is currently interrupting her and that her preferences are not very different from the current settings. A large difference would mean that she is not happy with the way her system is currently interrupting her and consequently prefers different settings.

The Table 30 and Table 31 below list the use cases that were employed for validating the cell phone and virtual secretary scenarios.

Case	CELL PHONE TEST CASES
1	Cell phone is with you and there is no incoming call
2	You have forgotten the cell phone in your office on a weekday and there is no incoming call
3	You have forgotten the cell phone at home on a weekday and the incoming call is a family emergency
4	You are speaking with someone on your cell phone, and there is an emergency incoming call from your family
5	You are listening to music on your cell phone with headphones on, and there is an incoming call.
6	Cell phone is with you, and you are speaking with someone (at a noise-sensitive location such as a meeting, hospital, etc.) and you get a casual call from a family member
7	Cell phone is with you, and you are speaking with someone (at a noise-sensitive location such as a meeting, hospital, etc.) and you get an emergency call from a family member
8	You are working in your office on a weekday, your cell phone is with you and you get a call from an unknown person
9	You are in an extremely noisy environment, your cell phone is with you and you get a call from your family
10	You are at a quiet place (not home or office), your cell phone is with you and you get a call from your family

Table 30. Use Cases for Validating the Cell Phone Scenario

Case	VIRTUAL SECRETARY TEST CASES
1	The professor is in her office and she is on the phone speaking with someone for about 10 minutes. Your visitor is a student.
2	The professor is in her office and is not on the phone, but has told you that she is busy 10 minutes ago. Your visitor is a very important person (e.g. dean)
3	The professor is out of town, and your visitor is one of her colleagues
4	The professor is in her office and is speaking to a student for close to 30 minutes, and your visitor is a student
5	The professor is in her office and is speaking to a student for close to 30 minutes, and your visitor is a very important person (e.g. dean)
6	The professor is in her office, she is not busy and there is no one with her at the moment. Your visitor is a student with an appointment
7	The professor has stepped out for about 10 minutes and your visitor is a student with an appointment
8	The professor has stepped out for more than half an hour and your visitor is a colleague
9	The professor is busy in her office with a student for more than a half hour, and your visitor is a student
10	The professor is in her office, is not busy and your visitor is a colleague

Table 31. Use Cases for Validating the Virtual Secretary Scenario

The cell phone scenario had 21 respondents selected at random while the virtual secretary had 18 respondents selected at random. Below we look at the use case comparison results for the virtual secretary scenario.

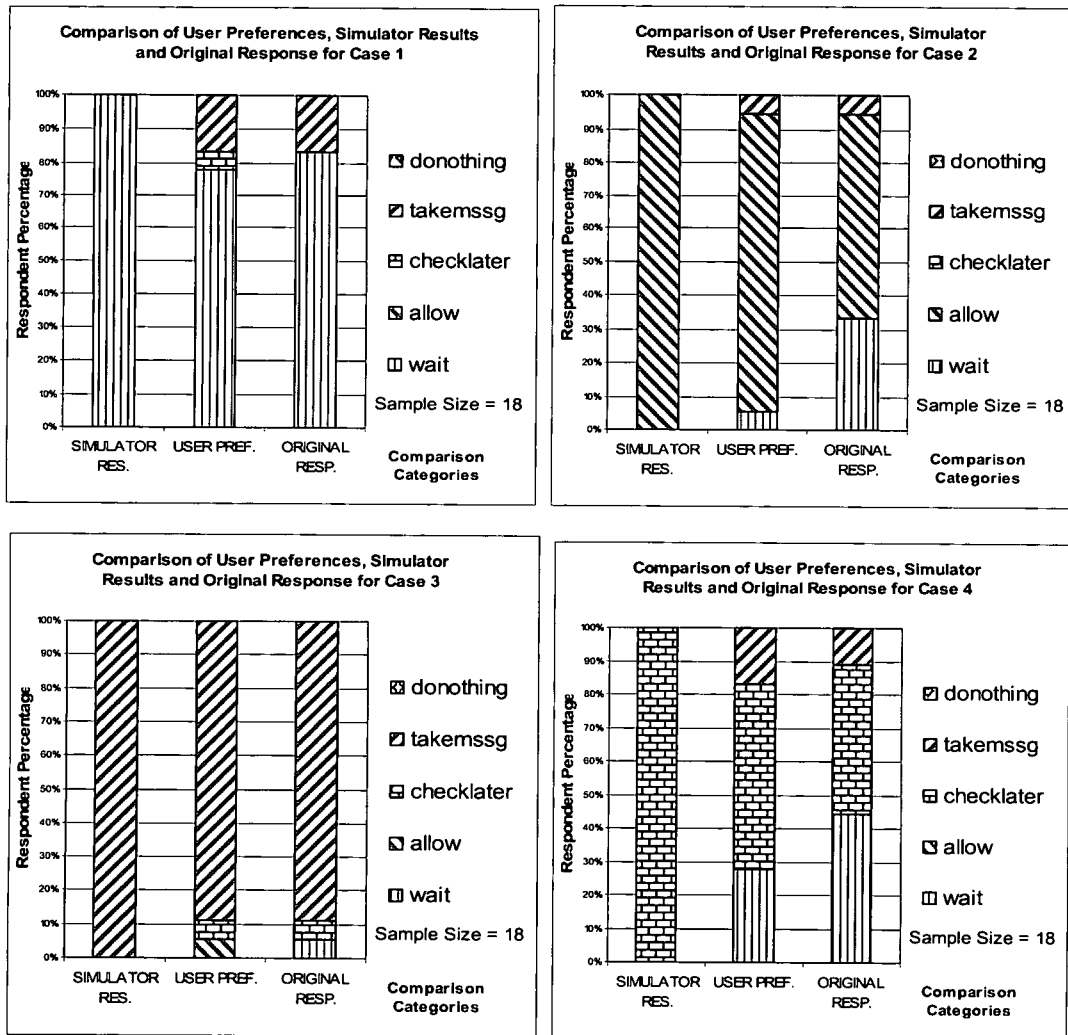


Figure 52. Use Case Comparisons for Virtual Secretary Scenario

We notice from these results in figures 52 and 53 that interruption handling for the virtual secretary scenario produced responses that were generally consistent with user preferences.

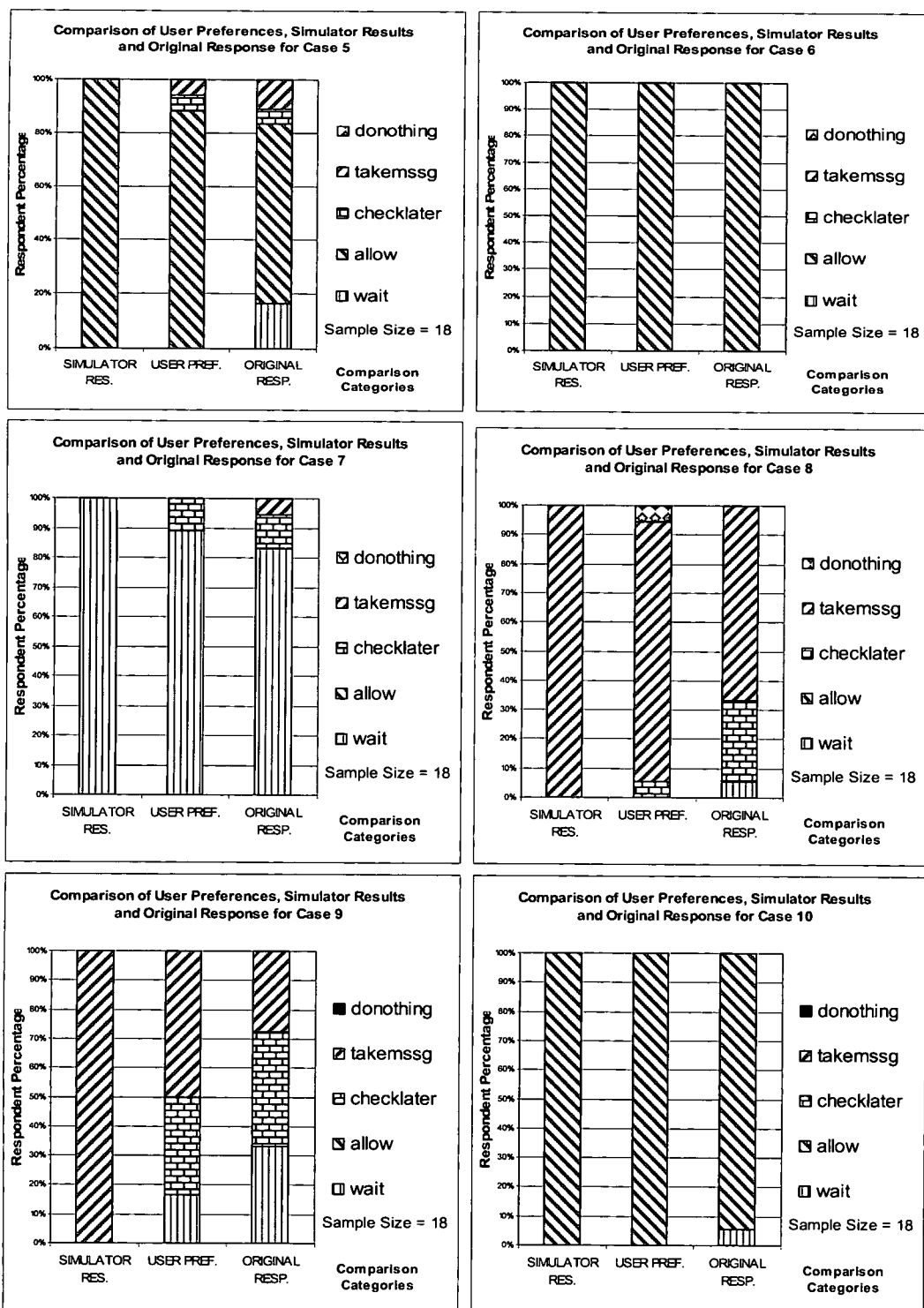


Figure 53. Use Case Comparisons for Virtual Secretary Scenario Continued

In many cases, over 80% consistency was observed (cases 1, 2, 3, 5, 6, 7, 8 and 10). In cases where the difference in simulator results and user preferences was large (cases 2 and 9), we can notice that alternate responses that users preferred (*checklater*, *takemssg*, *wait*) also reduce interruptions caused to the professor inside the office, thereby remaining consistent with the simulator results.

From these charts, we also notice that there is little difference in user preferences versus original responses (original responses are possible protocols that secretaries have to follow when responding to visitors). From field observations it was noticed that secretaries and office assistants very rarely follow visitor handling protocols (if any exist), and the handling process is based on factors such as the type of visitor, time of day and purpose.

In Figure 54 on the next page we look at results obtained for the cell phone use cases comparisons. The studies performed were conducted in exactly the same manner as those conducted for the Virtual Secretary scenario.

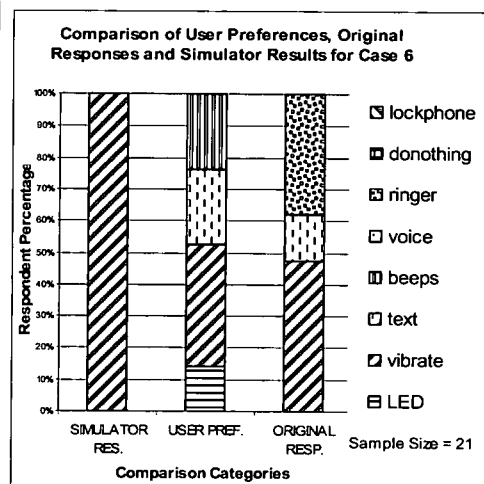
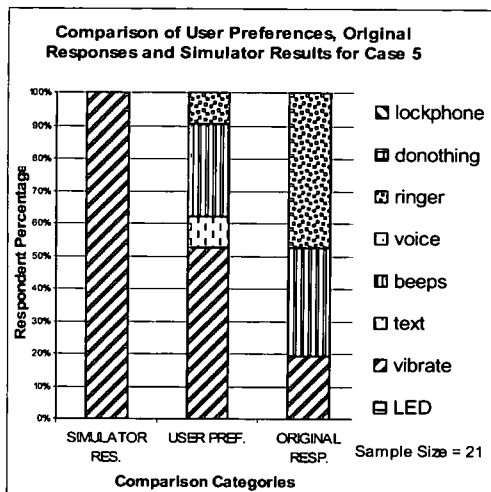
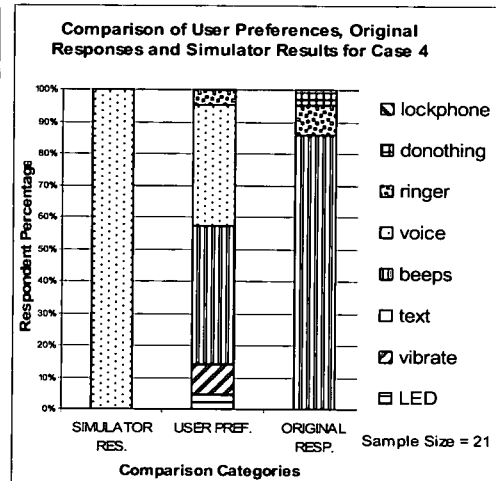
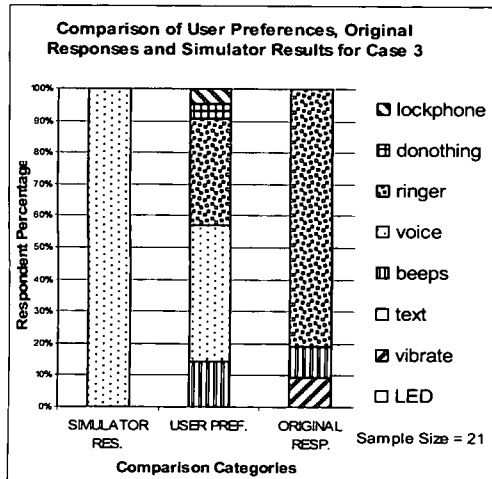
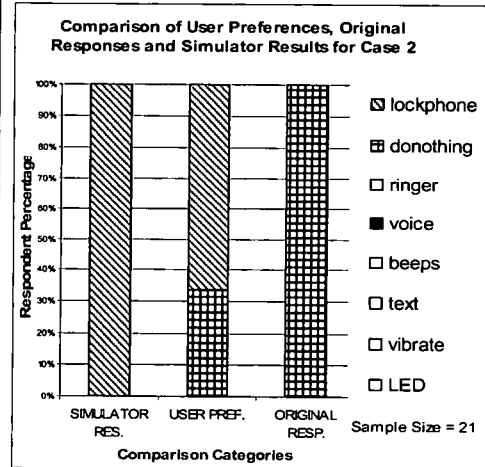
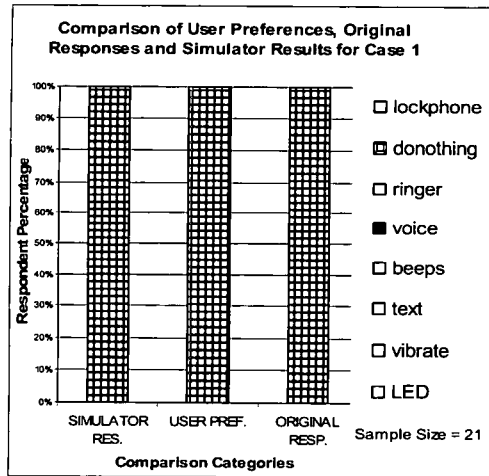


Figure 54. Use Case Comparisons for Cell Phone Scenario

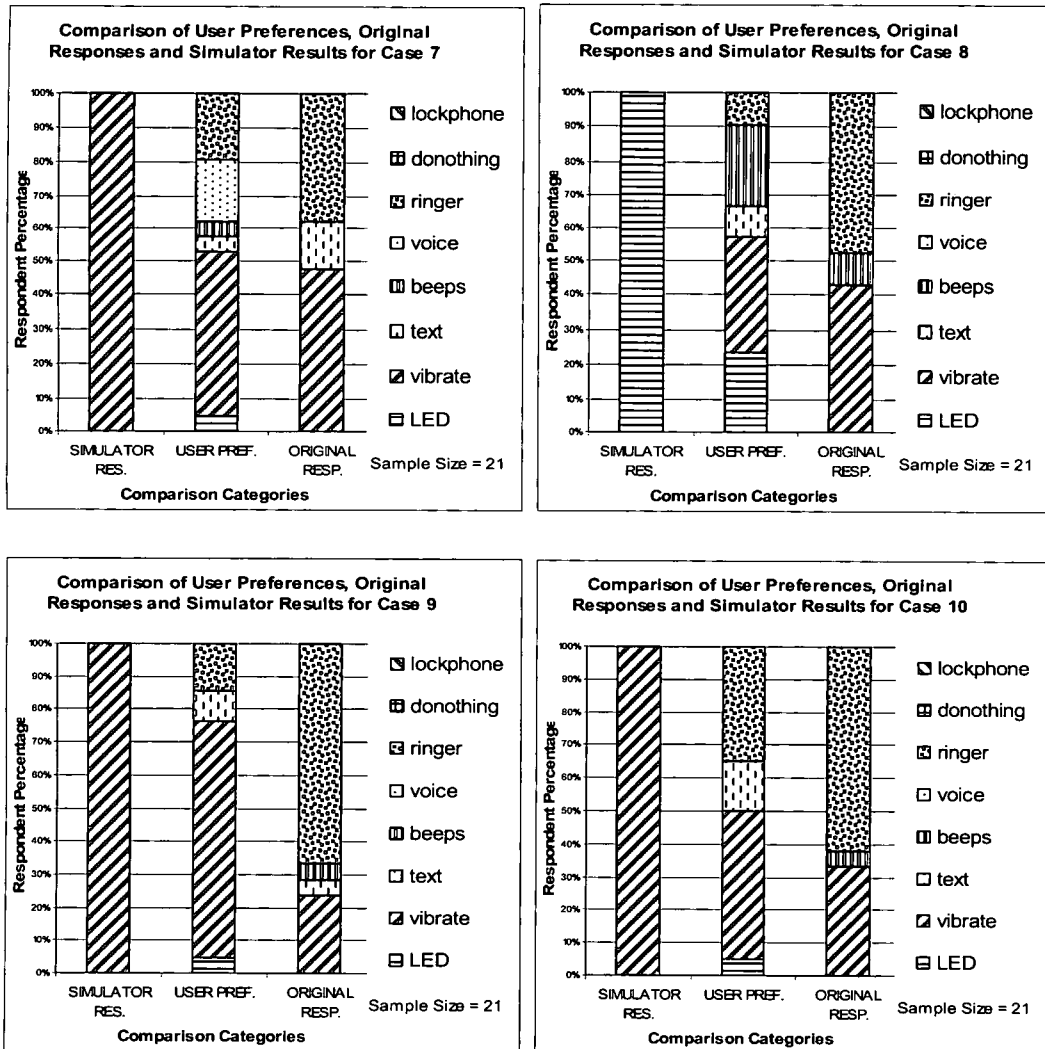


Figure 55. Use Case Comparisons for Cell Phone Scenario Continued

As with the virtual secretary scenario, the cell phone simulator results shown in figures 54 and 55 are generally consistent with user preferences. We notice that in four cases over 50% consistency was observed (cases 1, 2, 5, 7 and 9). In the remaining cases, at least a 40% consistency was noticed in simulator results and user preferences. The drastic reduction in percent consistency compared to the virtual secretary scenario is most

likely because of the following reason. Unlike the virtual secretary scenario where secretaries have to follow a general (sometimes implicit) guideline to handle visitors, cell phone users have various choices of interruption modes to choose from. Their preferences depend on their individual needs, situations and choices compared to the secretaries' preferences which are based not on their individual preferences, but on the interruptability of the professor inside the office. Hence the cell phone scenario is more likely to generate alternative responses to handling interruptions than the virtual secretary scenario. This can be clearly noticed in the charts above.

For example, consider case 6 where the user is at a noise sensitive location and is speaking with someone when she receives a casual call from her family. The simulator response was *vibrate*, while users preferred *vibrate*, *text* and *LED* as the alternate modes (all capable of reducing interruptions). It can also be noticed here that the default mode ringer doesn't occur in users preferences.

Thus the consistency of users' preferences, whether for passive notification modes for interruption sensitive situations (for example cases 5, 6, 8, and 10) or for active notification modes in cases of family or emergencies (for example cases 3 and 4) is maintained by the simulator results.

In spite of the successful validation it must also be noted here that the use of models and simulations for validation instead of actual prototypes had the following drawbacks.

- The experiments could not be run in real environments as could have been done with prototypes.

- The experiments did not duplicate the interruption properties such as timeliness, periodicity, etc. which would have been possible with prototypes.
- The users were not free to move about in their surroundings.
- Users had to be trained to use the simulators and models.
- Users did not have the patience or learning ability to perform the simulation activities and generate results by themselves.
- The validation tests required extensive intervention of the expert to aid the user in performing validation activities.
- Users sometimes did not understand simulator results which looked different from the proposed product.

Thus, in order to make end-user validation more effective and accurate, the models and simulations need to be converted into prototypes that can actually be used by end-users in their everyday environments.

In spite of these drawbacks it was felt based on the results obtained that user validation studies conducted here were effective enough to conclude that the interruption handling method used was successful and satisfactory to the users. In case of the cell phone scenario, it is definitely helpful to have technology that is aware of a user's situation and assists in reducing interruptions to her automatically. Surveys suggest (refer to the section on survey analysis) that users normally use ringer and vibrate as their default notification modes and have to remember to switch modes based on their situation. Surveys also suggest that users would like to see such technologies in their phones provided they are not too intrusive.

In case of the virtual secretary scenario, the idea is not to take away a human secretary's job but to determine whether machine learning and context awareness can be employed to provide automated gate keeping duties to reduce the burden on the already overworked secretaries. Results obtained here suggest that learning and awareness based techniques can indeed replicate gate keeping duties of secretaries accurately. In the next chapter, we look at various implementation issues to be considered when designing interruption aware context aware systems.

CHAPTER 9

IMPLEMENTATION ISSUES

In this chapter we look at issues that need to be considered for implementation and deployment of interruption aware systems such as the cell phone and virtual secretary scenarios. First we look at general implementation issues that become relevant during the implementation stages. These issues affect the choice of enabling technologies based on a variety of criteria. Then we look at various system design technologies that can be employed to implement interruption aware systems.

9.1. General Implementation Issues

There are a variety of general factors that need to be considered before we consider implementation issues related to enabling technologies when designing interruption aware systems. These factors are listed below:

- Cost
- Need/Market
- Implementation Complexity
- Privacy
- Security

- Stability
- Scalability
- Mobility
- Transparency
- Interference

These factors cover a variety of issues ranging from business, marketing and financial aspects, to system design. Cost and marketability are important issues due to the fact that interruption aware functionality is an uncommon property in cell phones and other commercial systems today. Despite user preferences obtained from surveys and studies, and testing of prototypes, models and simulations, there is no certainty that the product will perform according to expectations and will become popular in the real world. Thus making design decisions on factors such as cost, marketability, performance analysis or even the projected return on investment is a risky aspect in the development of such systems. Another aspect related to cost is feasibility and implementation complexity. This comes into the picture when the required effort in terms of cost, labor and time to add new services to existing systems conflicts with the returns expected. This is where many research prototypes, designs and projects fail to be converted to commercial products.

Other factors to be considered are mainly system related or human factors related issues such as scalability, mobility, privacy, transparency, and so on. Factors such as scalability and mobility are essential when the scenario is large and is deployed across multiple environments, cultures, or even user groups. Some of these factors cannot be benchmarked using models and simulations, and for this prototypes need to be field

tested. Issues such as privacy, transparency and interference are user related in that they are based on users' requirements and preferences. Users generally are in favor of technology for performing a variety of activities, but this preference changes quickly if the technologies actively pervade their personal spaces. Also, users are very concerned about privacy and sharing of personal information across the public domain. These issues are crucial in mobile, pervasive and wireless applications due to privacy and security concerns. Some of these issues while not technical in nature, directly affect the selection of enabling technologies for implementation. Below we look at various system design technologies that can be employed for implementation of interruption aware systems.

9.2. System Design Technologies

There are a variety of technologies that determine the implementation and development of interruption aware systems. For simplicity they have been categorized into four types, namely sensors, network infrastructure, computing systems and devices, and user environments.

9.2.1. Sensors

Selection of sensors for designing interruption aware systems is a vital task because the entire model for interruption handling, whether it is learning based or rule based depends on the attributes and attribute values that are chosen. These attributes and their values in turn depend on the sensors that have been selected. This is of course true if sensor based design is the approach used for interruption awareness, otherwise sensors and attributes are not very essential. For the purposes of this research, we can divide

sensors into two categories; hardware, and software. Table 32 shows a list of sensors along with some of their important parameters for consideration.

Sensor Type	Possible Attributes	Range/Power	Cost	Connectivity and weight
Hardware Sensors				
Light sensors	ambient light, indoors brightness, outdoors, etc.	~0-50 feet (0-25000lux) / 2-5 Volts	\$0.10 - \$50	Wired, wireless, analog, digital / 5 – 50gms
Accelerometers	motion, movement, vibration, physical state, etc.	+ or – 300g / 3-5 Volts	\$5 - \$20	Wired, wireless, analog, digital / 25gm – 1kg
Video cameras	facial expressions, gaze, behavior, presence, etc.	>1lux / CCD / VGA / 2-12 Volts	\$50 - \$1000	Wired, wireless, digital / 200g – 1kg
Microphones	speech, noise, music, decibel levels, etc.	~20-20KHz// 2-12 Volts	\$10 - \$300	Wired, wireless, analog, digital / 10 – 1500gm
Motion detectors	presence, single or multiple users, etc.	IR / Ultrasound / 5-12 Volts	\$5 - \$100	Wired, wireless, analog, digital / 50 – 250gm
Pressure sensors	pressed, occupied, hand gestures, etc.	~0-100psi / 2-12 Volts	\$0.50 - \$25	Wired, wireless, analog, digital / 10 – 100gm
RFID	location, activity, situation, etc.	100KHz – 1GHz / 1-12 Volts	> \$0.10	Wired, wireless, analog, digital / 1gm – 10Kg (including receivers)
GPS	Location	1.1GHz – 1.6GHz / 2-20 Volts	\$200 - \$1000	Wired, wireless, analog, digital / 500 – 2000gm
Environmental sensors (temp, humidity, etc.)	weather and other environmental conditions		\$5 - \$100	Wired, wireless, analog, digital / 10 – 2500gm
Software Sensors				
Event monitors	GUI events, schedules, notifications, errors, updates, etc.	Windows, Linux, Unix, Mac Operating System Monitors, C, Java, XML, and other programming environments, etc.		
Action listeners				
Loggers				
Agents, processes				

Table 32. Possible List of Sensors for Interruption Aware System Design

Hardware sensors include sensors such as light sensors, accelerometers, video cameras, microphone arrays, motion detectors, pressure sensors, and technologies such as RFID and GPS. They can be embedded in the system or externally connected. They can

also reside in the environment depending on the scenario. The key properties to consider in case of hardware sensors are power, bandwidth, range and robustness.

Power is an issue because many of these sensors may be embedded in environments with inadequate access or connectivity to power sources. In such cases they may have to carry their own power supplies which can cause additional burden and weight issues. In case of independent power supplies issues such as life of the supply, energy provided, replacement, etc. need to be considered especially in scenarios where the sensors locations are difficult to access, repair and maintain frequently.

Bandwidth is also a key issue because it determines how much and what kind of data is exchanged between the sensor and the main system. In many cases, sensors provide a limited variety of information that needs to be processed by the main system. This may be digital or analog in nature and in some cases may contain extra bits pertaining to identification, security, interface protocols, etc. Some sensors may transmit continuously, some may be periodic in nature, while others may transmit only when pinged by the main system or the receiver. The nature of data exchange is also dependent on the network infrastructure, whether it is wireless or wired and as we will see in the next section, there are a variety of possibilities.

Range determines the physical dimensions of the scenario environment and can be as small as a few meters, to as large as few kilometers. In addition to the type of sensor used, the power supply provided also determines the range to which sensors can function and transmit. GPS and RFID sensors naturally have greater operating ranges than say light sensors or microphone arrays.

Robustness is essential because many of today's environments are pervasive in nature and the sensors that are employed are embedded in various locations and objects residing in diverse environmental conditions. The main factors that affect the robustness of sensors are the operating temperature, resilience to motion, vibration, humidity, fire, and so on.

Software sensors on the other hand are relatively easy to deploy and the only relevant issues are access to the system services or data, rights and restrictions, installation, and sampling. The first few issues depend on the software environment and operating system in place while sampling is mainly concerned with how often the data is to be mined. Again, this could be event controlled, periodic, random or continuous in nature. In addition to the basic loggers, monitors and listeners, macro level processes and agents can be developed to act autonomously and mine for information independent of the user and her activities.

Most of the sensors discussed so far are modality specific and provide very little information other than what they are designed for. For example, light sensors can only provide information about the surrounding brightness, darkness or ambient light. This limits the knowledge learned from such sensors because an intermediate layer that fuses this sensor information to generate or modify knowledge needs to be created. For example, combining light, temperature and humidity sensors could determine whether it is a good day to play outside.

RFID sensors on the other hand provide location or activity specific information, and are not modality dependent. Their main advantage is that they provide data about the place or object they are embedded in, and this place or object may be something that the

user is situated in or is using. For example, RFID sensors placed in kitchen cabinet doors can provide information about whether the doors are being opened or shut. This could indicate the user's presence in that area and also shed light on her activity at that moment. Another advantage of using RFID is that they operate in transmitter-receiver pairs and depending on the range, they can make a scenario highly pervasive and mobile. This allows them to provide considerable awareness of an activity or location. They are also cheaply available and can be installed easily. Furthermore they are small and do not clutter the user's environment. Processing RFID data is simple because it is mainly information related to identification and any other bits attached to it.

Figure 56 illustrates typical sensor connectivity for an interruption aware system with a central computer based information processing system.

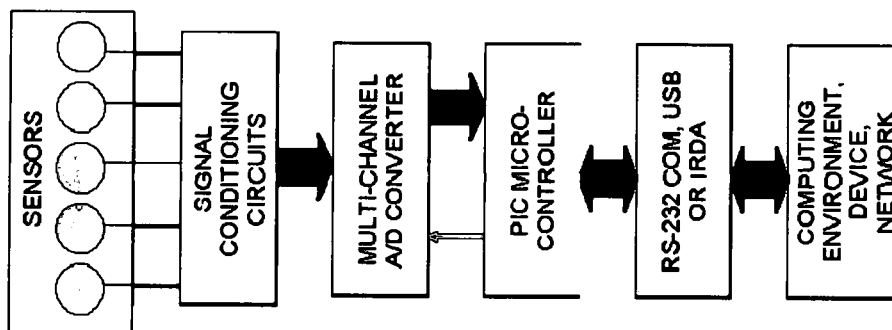


Figure 56. Typical Sensor Connectivity in an Interruption Aware System

The figure shows how different sensors can be connected via A/D converters to mobile or desktop PCs and other electronic devices such as cell phones, PDAs, etc. In the following section we will see more about the connectivity issues and network infrastructure required to set up such environments.

9.2.2. Network Infrastructure

There are a variety of enabling technologies available for setting up network infrastructure for interruption aware systems. They can be broadly categorized into wireless, wired and hybrid technologies respectively. Figure 57 shows an architecture that uses a combination of wired and wireless network infrastructures to provide context awareness and other human centric properties, and potentially interruption awareness also.

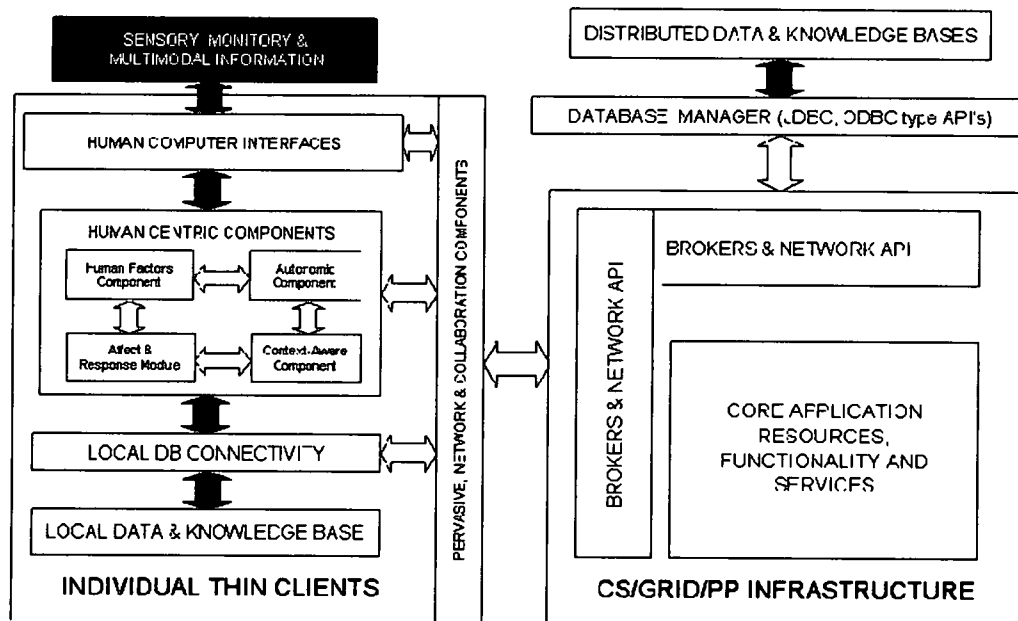


Figure 57. An Example Pervasive and Distributed Environment with Mixed Network Infrastructure

As the figure shows, a variety of systems (thin clients) may be connected to a centralized grid, to a LAN-based system, or peer-to-peer network infrastructure. The thin clients may be mobile and wireless users, desktop PCs, other networks, or even other sensor-networks. Since sensor information is typically high bandwidth and high volume in nature, local database connectivity and sensor information processing capabilities are

required. As shown, most human centric functionality such as awareness, affect and emotion, etc. is provided locally to prevent frequent 'bandwidth-hogging'. The centralized infrastructure provides all the main application resources, functionality and services for each connected client.

Given the type of scenarios and applications in today's aware systems, wireless technologies are understandably more popular. Some of the common technologies in use today are Bluetooth, IrDA, Wireless USB, WiFi and RFID. Wireless infrastructures are employed in most pervasive, distributed and aware environments because they offer high flexibility, mobility, speed, transparency and pervasiveness. They also provide effective context and situation awareness because they allow sensors and devices to be thoroughly embedded in the environment. Some technologies like IrDA and Bluetooth may be used for smaller scenarios while others like WiFi and RFID can be used for geographically dispersed environments. Wireless technologies have a few disadvantages though. For example they are not as reliable as wired networks due to their inherent broadcast and interference properties. Owing to this they are also prone to privacy infringement and security flaws because the data that is broadcast is potentially open for anyone to access and decode. Other primary challenges for wireless technologies are power, range and connectivity. Some of these issues were discussed in the section on sensors and they apply to network infrastructures as well.

Wired network infrastructures provide another alternative to connecting sensors and computing environments together to form pervasive and distributed environments. While initial developments in such environments were mainly wired in nature, the growing popularity and effectiveness of wireless technologies has relegated wired

networks to home and office environments for activities such as LAN connectivity, server-client setups, etc. Hence they are used mainly in smart home and office scenarios where they can be integrated with the existing electrical setups. The advantage of wired network infrastructures is that they provide high speeds, connectivity and reliability. On the other hand the cables, routers, switches and a host of network components make them highly intrusive. Other issues with wired networks are range, environmental factors such as moisture, heat, etc., and maintenance issues like upgrades, replacements, repair and debugging. Due to the reduced ability to provide embedded sensing and data capture capabilities they are not very effective in context and situation awareness.

A third approach is the use of hybrid and mixed network infrastructures as shown in Figure 57 that combine both wired and wireless technologies to provide a truly pervasive and distributed environment. Systems built using these technologies usually employ wired infrastructure for basic tasks such as setting up LANs and storage repositories, while the sensing and information retrieval is entirely wireless in nature to allow maximum embedded capability, high range and mobility. The only disadvantage in this approach is the cost and implementation complexity due to the use of different kinds of equipment, protocols and interfaces.

9.2.3. Computing Systems and Devices

For the purposes of this research we have defined the term 'system' as any type of computer or electronic device whether single user or distributed in nature that has the capability to process sensor information and respond accordingly. Sensors and other data

gathering devices may or may not be embedded in such a system. Examples of some systems that are used for awareness and other sensor based activities are:

- Personal computers (home, office, desktop, mobile)
- Workstations, servers and compute intensive systems
- Control stations (manufacturing and assembly plants)
- Data storage devices (tapes, disk drives)
- Printers, faxes, copiers, and other office machinery
- Microprocessors and microcontrollers

All of these systems in some form or another make up an aware computing environment (whether it is single user, or distributed). They perform tasks such as processing sensor information, providing a host of applications and services, or simply storing a variety of information. In addition to these 'core' systems, there are a variety of electronic devices, commercial products and objects that can also be part of an aware computing environment. Some examples are as follows:

- Cell phones, PDAs and handheld mobile devices
- Household appliances such as microwaves, ceiling fans, stereos, refrigerators, coffee makers, etc.
- Business/office equipment like printers, faxes, vending machines, staplers, etc.
- Objects like kitchen cabinets, walls, crash test dummies, F-16 cockpits, clothing, etc.

These serve as sensor hosts or objects which can be embedded with information gathering sensors. They may be wired or wirelessly connected to the core computing infrastructure as shown in figure 57. The main requirements of such systems are:

- Operating systems and APIs (application, services, network, device protocols and interfaces).
- Storage (RAM, secondary storage, fast caches, online and offline storage)
- Processing power (microcontrollers, full fledged processors, multi-threading, multiprocessing capabilities).
- Human computer interfaces (displays, audio-visual equipment, multimodal interfaces).
- Software applications, resources and services.
- Privacy, access management and security infrastructure.

We see that a variety of systems, components and software entities need to be integrated effectively to yield a good backbone from which the entire aware environment can be built by adding network infrastructure, sensors and other components. In the next section we will see possible user environments where such systems can be employed.

9.2.4. User Environments

The choice of enabling technologies such as sensors, network infrastructure, computing systems and devices, information storage, etc. for any given scenario depends a lot on the user environment. User environments can be categorized broadly into four types based on scenarios as follows:

- Home and Office Environments – involve mostly indoor environments with artificial lighting that is not excessively bright. The noise levels for such environments may generally range from 0 to about 90db. Since these environments are mostly closed spaces, the types of sensors and network infrastructure employed would have low to medium transmission range. They would also have low to medium bandwidth due to

the type of information that occurs in home and office environments. Power supplies for such environments would essentially come from wall outlets and lower power supplies. Infrastructure such as devices, sensors and network components for such environments is generally not expensive due to their commercial availability. Awareness in such environments is mostly activity based rather than location based due to the closed nature of the scenarios.

- **Outdoors and Public Spaces** – these environments generally refer to cities, towns and villages, highways and other outdoor spaces such as malls, parks, streets, etc. Due to the vast geographic nature of outdoor spaces, it is necessary to constrain the scenarios carefully. Important issues while designing for such scenarios are power (amount and supply), range and robustness. Light levels may be very bright in some instances and noise levels may vary from 10 to about 120db depending on the location of the scenario. Sensors and other components for such scenarios need to have protection against weather (rain, heat, humidity, etc.) as well as dirt, dust, etc. Depending on the scenario limitations, the operating ranges may vary from a few feet to thousands of kilometers. Again, this decides whether designers want to go for wireless, wired or hybrid environments. Power sources also need to be determined. For example cell phone based sensors may derive power from the batteries themselves but that may lessen the talk time or standby time. Awareness in such environments may be a combination of location and activity due to the nature of the scenarios.
- **Healthcare Environments** – these environments were categorized separately due to the differences in the types of scenarios and enabling technologies they employ. Such environments are mostly indoors and closed spaces while there may be some

applications requiring continuous body worn sensor data retrieval in users' natural environments. Sensors mainly gather information about bio-chemical events occurring in users' bodies. These sensors need to be highly accurate and reliable, while having long operating life and power supplies. The network infrastructure may be wired or wireless. Most sensor and network components have low ranges due to the proximity of the information processing units.

- Military Environments – a lot of research is currently being conducted in developing integrated command and control environments that are intelligent, aware and distributed in nature. Such environments are extremely large and complex with possibly hundreds of users; a variety of computing platforms, sensors and expert systems; and provide a host of services ranging from decision making and knowledge management to collaboration. Sensors and other infrastructure need to be very accurate and highly resilient to environmental conditions. Operating tolerances for such devices and systems are usually very large and they are able to function in diverse physical environments.

The following list of factors was determined based on the type of environment, affects the choice of enabling technologies employed.

- Heat and operating temperature
- Humidity, water, dirt
- Bio-chemical limitations
- Light intensity, noise level
- Movement and vibrations
- Sensor weight and dimensions

- Object supportability
- Available power sources
- Static/magnetic shielding
- Sensor output ranges
- Environmental interference and clutter

We can notice that different environments and scenarios focus on satisfying different factors mentioned above. Some give importance to physical environmental conditions while others focus on issues like the availability of reliable power.

9.3. Attribute Alternatives

The current list of attributes and their values selected for both the cell phone and virtual secretary scenarios work effectively in a simulation setting. In other words, the training datasets that are created from these attribute value combinations (without the user of actual sensors) represent to a large degree actual user interruptability and situations. This makes interruption awareness of the models that are generated very effective to new instances and situations. However, this performance is limited to simulation and modeling, and since we assume that the attributes selected accurately reflect the sensors and their values, the actual implementation results may depend on the accuracy of the sensors themselves.

For example, it was assumed that noise levels can be categorized into four types based on EPA noise charts. In reality this may or may not be the case depending on the scenario and the sensor sensitivity. In another example, it was also assumed that recognition algorithms can accurately differentiate between speech, noise and music.

Currently, there is considerable research activity in this area but the results are not good enough to incorporate such services in a commercially marketable product.

While such assumptions were necessary to simplify the scenarios, other complex techniques such as activity inference can also be employed to eliminate these assumptions. There are a few tradeoffs in doing so however. These tradeoffs are listed below.

- The process of activity inference is itself a very complex task since human activity is an extremely large combinational set of situations, locations, interaction, events and behavior.
- It cannot be performed effectively without the use of a large number of sensors and using a lot of sensors has many disadvantages
- Activity inference may cause too much environmental clutter due to the large number of sensors which makes it disruptive to user activities.
- It involves increased processing complexity since a larger number of sensors imply a greater number of attributes and attribute values.
- It also involves increased processing and storage requirements due to the large training dataset and model sizes. Unsupervised learning may reduce the storage requirements but it also reduces the overall accuracy and effectiveness.

RFID is a main component of activity inference based awareness. It was mentioned before that RFID provides object, location and activity specific information, rather than modality specific information. It can also be embedded completely in users' environments. Due to these abilities it allows for recognition of specific activities in

combination with other simple sensors, such as opening and closing kitchen cabinets, sitting at work desk, sleeping, going to the bathroom, etc.

Thus the set of attributes for the cell phone scenario can be changed from ones that detect interruptability and busyness to ones that detect activity. An example is shown as follows:

```
{decibel, soundtype, motion, appointment, purpose, caller,...}, changes to  
{location, roomtype, devicetype, doortype, caller, purpose, appointment,...}
```

We see that the first set of attributes primarily provide modality specific information and are used to detect how interruptible or busy the user is, while the second set of attributes are a combination of RFID and simple sensors, and provide information about user situations and activities directly without having to process the sensor information. We see that while attribute selection based on activity inference can be very specific and accurate, it also requires more sensors and attributes to achieve the level of accuracy. On the other hand, if we are only interested in interruptability and busyness, simple sensors can also be equally effective. In this work, RFID based attributes are sparingly used because the approach is to avoid activity inference and focus more on recognition of interruptability and busyness. This enabled the scenarios, attributes and training datasets to be simpler and the learning model was also easier to understand.

The next section concludes with a summary of the work done, a list of main contributions of this research, and a discussion of possible directions and future work in the area of interruption aware systems design.

CHAPTER 10

CONCLUSIONS AND FUTURE WORK

This section concludes the dissertation by presenting a summary of the work done on interruption handling and development of the scenarios. Here we will also look at the main contributions of this research towards the general area of human centered design, and more specifically in the areas of context-aware systems, and interruption handling. Possible research directions and open issues that need to be addressed in these areas are also discussed in this section.

10.1. Summary and Conclusions

To summarize, this research begins by viewing system generated interruption as a significant problem in the design of existing and future human centric systems. A specific area in human centric systems namely context-awareness is chosen to narrow down the scope of the problem. Fundamental concepts are presented to understand terminology, principles and relevant issues in the areas of context aware systems, human centric systems and user interruption. The work then focuses on the resolution of system generated user interruption by first looking at existing solutions, proposing a taxonomy for the existing solutions, and presenting a new approach to solving the problem. This

solution is based on user involvement throughout the design process and employs a learning based approach that integrates user input for generating models that resolve interruptions. The proposed solution is then simulated using two popular and highly interruption prone scenarios namely cell phones and virtual secretaries. The performance of the simulators and learning models is tested in terms of speed, classification rate, and size. User validation studies are then conducted to ascertain whether the new method is effective in resolving interruptions and whether users are satisfied by it or not. Important implementation issues are presented in order to allow designers to move from simulation to development stage. Finally, the dissertation concludes with discussion about possible directions for future work.

After reviewing the considerable literature on both interruption handling methods, and context aware systems, it was felt that many of these systems do not go beyond the prototype stage for reasons ranging from design cost and implementation issues, to user satisfaction. While factors such as design cost and implementation issues can be addressed by careful selection of enabling technologies, user satisfaction can only be achieved by effective design. One way of effectively designing for the user is by incorporating user needs and preferences [Cooper, 2004] [Shneiderman, 1997] [Norman, 1986] early in the design process. This research follows this approach by eliciting information from users via surveys and field studies and incorporating the results directly in designing the interruption handling models. By doing this the simulators are able to accurately portray user's preferences and interrupt her in an appropriate manner.

The use of learning methods also allows the models to be dynamic in nature and respond according to situation. Incremental learning can increase this dynamic ability by

making the models rely less on historical evidence and focus more on real time situations, thus reducing the response time and increasing speed. Therefore, at a simulation level, the combination of information elicitation and learning based approaches proves to be highly effective in handling interruptions in a situation specific manner reducing their disruptive effects on user's activities. The next section looks at the main contributions of this work in the areas of human centric systems design and interruption handling.

10.2. Main Contributions

The work done in this dissertation focuses on several important topics ranging from issues related to human centeredness, context-aware systems, and interruption handling to user involvement in system design via usability engineering techniques or direct participation. The following is a summary of the major contributions this dissertation provides in the aforementioned areas.

- Terminology, frameworks and methodologies are presented that provide a thorough understanding of the fundamentals in the areas of human centeredness, human centric design and human systems integration, based on the study of various research efforts, relevant design issues and current limitations. The framework for human centric design captures major human centric properties along with their design issues and enabling technologies. A unique outlook is also presented that divides human centric system design into categories based on an end-user perspective, an environmental perspective, and a systems perspective. This discussion will allow designers to clearly focus on specific design goals and requirements from an end-user perspective.

- A taxonomy and design process is developed in order to understand the principles of context-awareness and context-aware systems. The taxonomy provides a deeper understanding of how knowledge is generated from raw low level data. This knowledge may be contextual, situational, location specific, activity based, or even preference related. It shows the different levels of knowledge ranging from low level information which is simply categorized data all the way to meaningful information that may have been generated by combining various categories, or various lower level information pieces. A generic context aware design process is also presented which illustrates how the raw data is converted to meaningful knowledge.
- A system design process is proposed that incorporates user participation in various design stages via direct involvement or via information elicitation. This process combines principles from Interaction Design and Participatory Design in order to incorporate human centric properties in systems. Both these areas are essential in developing human centric systems with properties that closely reflect desired goals, requirements and preferences. This process is used as a guideline for developing simulations for two scenarios in this work and can also be used for developing other types of dynamic and aware systems.
- The focus of this research is on resolving system generated user interruption in human centric and context aware systems. This is especially important given the growing popularity of such systems with their pervasive, dynamic and situation-aware functionality in applications like mobile devices, smart homes and virtual assistants. System generated interruption, its causes, characteristics, and categories are discussed in detail to provide a better understanding of the problem. A new

taxonomy of existing solutions is presented with discussion about the different approaches to resolving interruptions. This taxonomy is essential for determining what solutions exist for handling interruptions, what is lacking currently and what approaches are best suited for specific scenarios and domains.

- In order to resolve user interruption, a method that handles interruptions based on user activities and situations is presented. This method employs machine learning techniques to gather information about user contexts in order to handle system generated interruptions in an appropriate manner. The method proposed is three fold, in that it employs pre design and post design information elicitation stages that feed into the learning model development stage in order to refine the design based on user input. The information elicitation stages are currently manual in nature and employ surveys, field studies and experience sampling to gather information and build the learning models in the absence of sensors. These can be automated to directly process the information entered by the user to generate models.
- The proposed method is validated by simulating two popular and highly interruption prone scenarios namely cell phones and virtual secretaries. Here we see how user preferences and input are accurately portrayed in the system functionality and responses to interruptions by analyzing information obtained from users. The proposed method is also evaluated using parameters such as size, accuracy and speed for possible implementation purposes.
- Implementation issues involved in the process of converting initial design simulations and models into actual prototypes or products are discussed. These issues are specific to the scenarios in consideration, but also may be generalized and expanded to

include other applications. These issues cover end-users, enabling technologies, infrastructure, environments and the underlying constraints that arise from each. Alternative approaches to the design of interruption aware systems and information elicitation are also discussed.

In the following sections we look at future work and extensions to this research. We also discuss possible research directions in the area of interruption aware systems and what is lacking in the field along with some open issues.

10.3. Future Work

In this section we look at possible extensions to this research effort in terms of prototyping the two scenarios, automating the information elicitation process and alternate approaches for interruption aware design.

10.3.1. Prototyping

Continuing with the discussion on the context/interruption aware design process, prototyping and validation appear as the final stages in the design of aware systems. Prototyping is essential to test the model, simulation, architecture or other aspects of the system design in a real world scenario. It enables designers to ascertain how well the initial designs work, what the limitations and drawbacks are, and also helps them go back in to the design process and modify aspects of the initial design that need refining. Prototypes may or may not have the entire set of functionality and services that the final system would have, since their main purpose is testing the initial design in realistic settings.

Figure 58 shows a typical prototype setup for the interruption aware cell phone scenario. We can see that the cell phone is connected to a laptop via a USB or serial port, or could also be a wireless connection in the form of Bluetooth, IrDA, or Wireless USB.

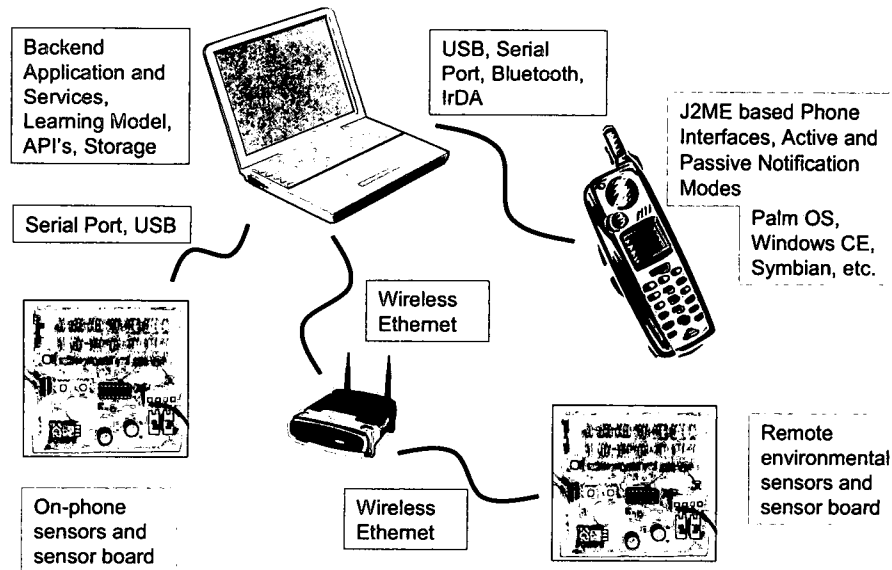


Figure 58. Initial Prototype Architecture of Interruption Aware Cell Phone Simulator

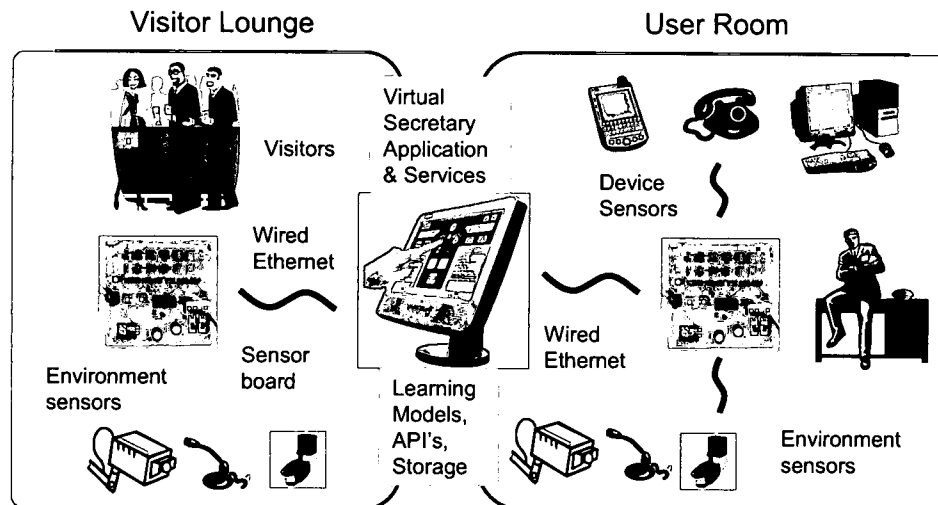


Figure 59. Initial Prototype Architecture of Interruption Aware Virtual Secretary Simulator

Notice the laptop and external sensor boards in the proposed prototype. This is because current cell phone technology does not have the processing power, memory or software environment flexibility to incorporate the entire interruption aware application, along with its services, sensor data storage and learning models.

A similar prototype is proposed for the virtual secretary scenario as shown in Figure 59. As the figure shows, a centralized computing system handles both the visitor side and the user side sensor information via wired or wireless networks. Visitors may use touch, text or GUI based input to interact with the secretary, or may also use natural speech and gestures depending on the type of sensors employed. The sensors on the user side are environment based as well as device based because the user may provide a lot of information on her interruptability or busyness via the devices she uses such as telephones, PDAs, or PCs.

We noticed in the section on user validation that the results of validation using models and simulations are generally effective, but not accurate due to the many drawbacks of using models and simulations. This is an important reason for developing prototypes. Prototyping is especially important in designs that incorporate user input in the form of surveys and field studies because, if end-users are satisfied with the prototypes, it means that their preferences and requirements have truly been incorporated in the design and that the design process was a success.

10.3.2. Automating Information Elicitation

Pre-design information elicitation is an essential activity that allows designers to gather insight on preferences and constraints as seen from end-user and scenario

perspectives. Here solutions to issues such as sensor and attribute selection, choice of enabling technologies, services to be offered, infrastructure limitations, etc. can be obtained. This stage is a common component of every engineering and software design process. In this research, this stage was taken a step further by directly using information from users, field studies and experience sampling to design the system. This process is similar to tailoring and participation.

While information elicitation and user participation are essential they increase the designer's workload since she now has to have expert knowledge on usability engineering techniques and analysis principles to elicit usable knowledge from the user input. One way to reduce this workload would be to automate some or all aspects of information elicitation. Table 33 illustrates different levels of automation possible in the process of information elicitation.


	Steps in Pre-Design Information Elicitation			Related Areas
				
Activity Type	Elicitation	Analysis	Integration	
Level 1	Manual	Manual	Manual	Usability and Human Factors engineering, Engineering Design
Level 2	Automated	Manual	Manual	Information Retrieval, Survey Coding
Level 3	Automated	Automated	Manual	Information Retrieval, Knowledge Management
Level 4	Automated	Automated	Automated	Information Retrieval, Knowledge Management, Decision Making and Support
Level 5	Man./Auto.	Man./Auto.	Man./Auto.	

Table 33. Different Levels in Pre-design Information Elicitation

Level 1 involves paper-pencil or web based methods for information elicitation. In this category designers create surveys, interview queries, field study criteria, and sampling forms on paper and perform all activities manually. User input may be paper based or device/web based and the designer has to manually convert free text, multiple choice and other types of responses into coded information for analysis. The analysis of this information is also manual in nature. It may employ qualitative, observation based methods or mathematical techniques. Integration involves a careful observation of the analysis results and conversion into rules and policies that are integrated into the system design. This research employs a level 1 approach in that all aspects of information elicitation are manual in nature.

Level 2 primarily involves online web based or device based information elicitation methods. In this category, designers automate the elicitation part of the process by converting the paper based techniques into PDA or web based techniques where users can fill out the surveys, or gather field information using computers and electronic devices, which provide designers with categorized and stored information. This information is further coded and formatted by automated coding mechanisms. Thus the designers only have to manually analyze and integrate the information in their designs. One disadvantage of this technique is that due to the automated coding process it restricts the content of the surveys and other elicitation techniques by making them structured. Input such as speech, free text and other qualitative data cannot be easily coded. On the other hand, there are many coding mechanisms that process structured queries effectively and reduce the burden on designers.

Level 3 involves automated elicitation and automated analysis methods. In this category, online web based or device based methods that are used for information elicitation are directly linked to coding mechanisms that convert user input into structured and coded data. This coded data is then analyzed using a variety of retrieval and mining algorithms to ascertain relationships and patterns in data. Again, the bottleneck here is the automated elicitation process which limits the type of data being input and hence the usefulness of mining algorithms is not fully realized. In this category, the results from the analysis are obtained in the form of rules and policies that can then be incorporated manually in the system designs.

Level 4 involves complete automation of the elicitation, analysis and integration processes. This works similar to a workflow process in that, each stage performs a set of activities and feeds into the next stage. Each stage has inputs and outputs that need to be in specific formats to be recognized and used in the stage. The elicitation and analysis work in a similar fashion as described above. The integration mainly involves taking the results obtained from analysis, such as rules, policies, conditions, variables and constraints, and using them to design certain aspects of the system. In this research, this process is manual and involves using the rules and policies to build training data to design the simulator models.

Level 5 involves automating any of the stages in information elicitation. The most common approaches are to automate the elicitation and analysis stages. These involve the use of data structuring, retrieval and mining algorithms to process user input and generate rules and policies for integration in the system design. This is the most flexible approach because it allows designers to gather different kinds of information ranging from

structured data to free speech and text. A variety of effective mining algorithms exist that can then work on such data to produce specific results such as rules, policies or even decisions.

10.3.3. Alternative Approaches

In the section on existing solutions we have seen that there are a few system driven approaches that can be employed to achieve interruption and context awareness, but they have some drawbacks.

For example, state and rule based techniques can be employed only for small scenarios with very few attributes in the attribute set, and a small population of possible instances. Furthermore, these techniques do not learn from existing information and as such will correctly classify only instances that already are well defined in the form of state transitions or rules and policies. For the cell phone and virtual secretary scenarios developing state diagrams, rules and policies is not recommended due to the large scenario space and number of possible instance combinations.

Other approaches such as instance based learning primarily work on history or memory-based classification and are computationally intensive. They not only require considerable memory to store previously occurring instances, but also require considerable processing time (and power) to compare each historical instance with a newly occurring one. These properties are not always available on resource constrained devices such as cell phones. Again, as with state and rule based techniques, instance based approaches can be employed for small to medium scenarios due to increased computational requirements.

Overall, static, rule based and instance based approaches are fixed approaches, i.e. they do not learn from previous or existing data and can only classify instances for which a solution exists (either as a state transition, rule or similarity measure).

On the other hand learning based techniques such as neural networks, self organizing maps, clustering, and decision trees (used in this research) offer greater advantages because they can learn from a small sample of historical evidence and classify new and previously unseen cases. Some of these techniques (decision trees) are static learners and others (clustering, self organizing maps, etc.) are incremental learners.

The main disadvantage of a decision tree learning approach such as that used in this work is that they are static learners and can only classify instances based on the model that was learned from historical evidence. Hence if the current situation or events change, decision trees are not capable of adapting and subsequently fail to maintain high accuracy.

Ideally, the cell phone and virtual secretary scenarios need to be adaptive in nature, i.e. gather information from situations in real time and respond accordingly. This ensures that they are suited towards individual user needs rather than a population of users (where the minority gets ignored). Furthermore, user preferences need to be automatically recognized rather than having users to intervene periodically (as is done now) to modify their preferences.

To achieve this incremental learning approaches need to be employed. Incremental learning approaches do not necessarily rely on historical evidence to classify new instances. Some approaches are partially incremental (i.e. use some form of history), and some are completely incremental (use no history whatsoever). Clustering techniques

and self organizing maps are examples of completely incremental techniques wherein incoming instances are used in real time to learn and categorize.

Below we look at incremental decision trees as a part of the proposed future work. The pseudo-codes below describe the original ID3 [Quinlan, 1986] decision tree algorithm (which is similar to the C4.5 algorithm used in this research), and a modified incremental decision tree algorithm ID5R [Utgoff, 1988]. ID5R modifies the ID3 algorithm by making it incremental in nature. Further explanation is provided below.

The Pseudo-code for ID3 is as follows:

1. If all class values of the set of instances are the same, create a leaf node and assign the class value to the leaf node.
2. Otherwise:
 - a. Create a non-leaf node.
 - b. Select an attribute with the highest Information Gain. The attribute selected may not be repeated in the same path during tree creation.
 - c. Assign the attribute to the created node.
 - d. For each value of the attribute at the node, create a branch.
 - e. At the terminating end of the branch, repeat procedure.

As we see here (and also from earlier discussions on decision trees) the model is created by determining the Information Gain of attributes at every node, and class responses are assigned only when all instances are of the same class. Once the nodes and leaves have been created, they stay fixed and are able to classify new and previously unseen instances. The problem arises when users modify the scenario definitions by changing attributes or attribute values, or even the original response to training instances. If this happens, the decision tree model cannot take the changes into account and will incorrectly classify any new instances that follow the modified scenario definitions. In order to make this model incremental and to allow changes to be incorporated into the tree model, the ID5R pseudo-code [Utgoff, 1988] is described below.

1. If tree is empty, create leaf node with class name as class of instance, and store instance at the node.
2. Otherwise, if tree ends with a leaf node and incoming instance is of same class, store instance at the node.
3. Otherwise:
 - a. Create a non-leaf node by choosing a random attribute to test at the node.
 - b. For each value of the selected attribute, update the total number of positive and negative instances.
 - c. If the non-leaf node contains an attribute that does not have the lowest Entropy
 - i. Determine attribute with the lowest Entropy.
 - ii. Transpose this attribute with the attribute currently at the root node.
 - iii. Follow the same process for all sub-tree nodes.
 - d. Recursively update the all sub-trees below the current node for the value that occurs in the instance description. If value does not occur in the sub tree, grow a new branch.

The basic principle of the ID5R algorithm is to update the counts of positive and negative instances at the nodes and restructure the original tree (and any sub-trees) by changing the attributes at the nodes based on the new entropy values. The reason for changing the attributes is that whenever a new instance comes in, the positive and negative count changes thereby changing the entropy for each attribute. If the current node no longer contains an attribute with the lowest entropy, it needs to be switched with one that does thereby causing the tree restructuring. This ensures that the principle of the decision tree (entropy or information gain based selection) remains intact while making it incremental in nature.

The only disadvantage of the incremental tree based method is that incoming instances and the resulting positive and negative counts need to be stored along with the tree structure. According to Utgoff [Utgoff, 1988] this is comparable to the original tree construction in terms of computational requirements. One way of storing the counts is in the form of a frequency table [Swere et al., 2005] as shown below in Table 34.

Class	A1	No. of instances	A2			A3		An	
			$V1_{a_2}$	$V2_{a_2}$	$V3_{a_2}$	$V1_{a_3}$	$V2_{a_3}$		$V1_{a_n}$	$V2_{a_n}$
C1	$V1_{a_1}$	100								
	$V2_{a_1}$	23								
C2	$V1_{a_1}$	345	0	345	0	221	124	173	172
	$V2_{a_1}$	232	12	78	142	232	0	116	118
C3	$V1_{a_1}$	45								
	$V2_{a_1}$	645								
....										
Cn	$V1_{a_1}$	3								
	$V2_{a_1}$	465								

Table 34. Frequency Table for Storing Decision Tree Instance Information

As the figure shows, the counts of each instance are stored with respect to its class value and attribute value. For example, 577 instances belonging to class C2, of which 345 contain attribute value $V1_{a_1}$, and 232 contain attribute value $V2_{a_1}$. Further breakdown for other attribute values for class C2 is shown in the table. Such a table can be stored along with the decision tree model for purposes of calculating entropy at each node every time a new instance comes in.

Incremental learning for interruption handling is truly beneficial in cell phone scenarios because of the constantly changing user preferences and situations. Human activity is inherently uncertain and incremental learning can provide added effectiveness which cannot be achieved with static methods. In the next section we will look at possible directions for research in the areas studied in this dissertation.

10.4. Possible Directions

It was noticed from the discussions on existing solutions for interruption handling that currently there is a considerable lack of research in this area. Interruption handling has often been viewed as a secondary issue, almost as a constraint that is generated when providing functionality and services to a system, rather than as an independent issue with its own requirements and needs. It was also noticed from the ideas and solutions discussed in this work that the problem of system generated user interruption has multiple causes and origins. It is therefore not sufficient to address interruption as a subset of another issue such as UI design and neither is it sufficient to address any one particular issue in interruption design and overlook others. We need to look at issues starting from the top level such as incorporating human centeredness in design, all the way down to issues at the lowest level such as software engineering and design.

At the topmost level, issues in the areas of human centric design, human systems integration and human computer interaction need to be addressed. For example Norman and Shneiderman [Norman, 1986] [Shneiderman, 1997] discuss usability principles in interface design and address a variety of human computer interaction issues that enable software applications to effectively interact with their users. In a similar manner, we need to look at functionality, services, properties and other aspects of system design that will make them more human centric and less prone to interruptions. We also need to address end user perspectives by considering human factors issues such as user response, psychology, social and cultural preferences, and so on. For example, users from certain cultures may be more sensitive to interruptions than others and a system designed for

global use must incorporate such preferences. Other aspects of human centeredness such as surrounding environmental conditions, situation and so on also need to be incorporated.

The middle level needs to include various domain and application specific issues such as mobility, awareness, pervasiveness, location and intelligence. Addressing these issues is necessary to make the system more suited to the domain, to the surrounding environment, and to the types of users that will use the system. For example, virtual tour guides that switch content dynamically based on location need to be aware not to disrupt user's activities. The growing popularity of intelligent systems, animated avatars and other dynamic applications has made it essential for designers to incorporate interruption handling functionality in such systems. More research is therefore necessary to understand the interaction between users and such systems with an eye on their surroundings (as well as other users around them).

The lowest level comprises of issues that are mainly system, design or infrastructure related. As figure 2 suggests, these issues are directly dependent on the top level issues. We noticed in the section on existing solutions that there are a few approaches for interruption handling, but they are either based on training users themselves, or are application specific and look at notifications rather than interruptions. Thus based on figure 7, more research is required on the development of techniques, algorithms and methodologies for interruption handling. While static, rule or case based techniques work accurately for many cases they are scenario specific and do not address multi-domain problems associated with interruption handling. There is also very little research in modality based interruption which focuses on using specific interrupting modalities for different kinds of notifications. While research on learning based methods

for interruption handling exists, it is again application specific and not scalable across domains. Furthermore some of this research employs learning based methods for provision of services and functionality rather than for direct interruption handling. Unsupervised learning techniques can be highly advantageous for interruption handling because they do not need to be trained, and hence can be applicable to different kinds of users, situations and surroundings unlike static or supervised learning methods. Unfortunately unsupervised learning techniques become exponentially complex as the size of scenario and its requirements increase, thus compromising their accuracy and speed in the process, factors which are essential for interruption handling.

Another possible area of research is the manipulation of interruption characteristics such as duration, content, timing and complexity in order to handle them more effectively. This could be done by the introduction of rules and policies or by learning based methods in combination with awareness techniques. This would enable systems to provide truly dynamic interruption handling that is not only sensitive to users and situations but also takes into account properties of interruptions themselves.

The inclusion of user input directly in system design via participatory design techniques, tailoring or direct manipulation is also an area of interest for interruption handling. This will enable user preferences, requirements and needs to be more accurately incorporated in system design leading to fewer inconsistencies and disruptions in user activity. While software and engineering design processes do employ usability techniques and user studies to gather information, this information is qualitative and in the process of analysis some of it gets lost. This approach is applied in post-design stages also where field and usability studies try to ascertain whether the system is performing in

accordance with the desired goals. Again, this process provides qualitative information that cannot be used to quickly modify the design. One area of interest is post-design information elicitation where users can set their preferences and choices by directly modifying the designed system to suit their needs. This works best when systems properties are incorporated based on statistical analysis.

APPENDICES

APPENDIX A

GENERAL SCENARIO SURVEY

I. General Short Answer Questions

1. Describe in a few words various activities you perform at work with the level of importance (priority) attached to each activity [you may use low, medium, high, none as possible levels]. Also include any non-work related activities you may perform such as breaks, visits, lunches, chat, etc.
2. Describe the devices you use to perform various activities during a random work day (work, interaction, entertainment and pass-time). Some examples are PC, laptop, intercom, cell phone, CD player, etc. Include details of the device used if possible.
3. Describe the level of experience you have with the following devices and software programs [you may use low, medium, high, none as possible levels]. Also describe activities you perform with them. [reading/writing documents, movies, surfing, chat, conversation, visual art, etc.]

Personal Computer (Laptop):

Cell Phone (Handheld):

Landline (Intercom):

Software Programs (Chatting and Instant Messaging, Web Browsers, MS Word/PowerPoint and other document related software, Microsoft Windows):

4. List various responses you may have on being interrupted by a person, or by expected/unexpected events in your computer/phone/other device [for example tremendously destructive rage at losing valuable data]. Interruptions can occur at any time; during work, leisure, night, day, etc. Responses can be as wide-ranging as the thought of sledge-hammering your PC after a crash or overwhelming joy after receiving an instant message; in an instant.

I. Mobile Phone Related Questions

1. What kind of mobile communication device do you use? List name and model number.
 - a. Mobile phone [_____]
 - b. Handheld [_____]
 - c. Smart phone [_____]
 - d. None
2. What activities do you perform using your mobile device?
 - a. Make and receive phone calls.
 - b. Surf the web (check email, etc.).
 - c. Send and receive SMS messages.
 - d. Store and update information (addresses, numbers, emails, etc.)
 - e. Perform calendaring and scheduling (make schedules, reminders, notes, etc.)
 - f. Document-related activities (word, powerpoint, excel, etc.)
3. What kind of notification modes does the device have?
 - a. Voice
 - b. Beeps
 - c. Vibration
 - d. LED (light)
 - e. Other (screen display, flashing icons on screen, animation, music, etc.) [_____]
4. How many times a day do you use your device (to call or receive calls only)?
 - a. 0 – 5
 - b. 6 – 10
 - c. 11 and up
5. How many times a day do you use your device for other purposes (see question 2)?
 - a. 0 – 5
 - b. 6 – 10
 - c. 11 – 20
 - d. 20 and up
6. Where is your device located while in use?
 - a. In shirt/top pocket.
 - b. In hand.
 - c. In jeans/skirt pocket.
 - d. On beltclip.
 - e. Other (on desk, in suitcase, in purse, etc.) [_____]
7. When your device is not in use, how far away do you keep it from you (across the room, different room, don't take it with you, etc.)?

8. Some activities you do when your device is with you.
 - a. Work
 - b. Exercise
 - c. Travel
 - d. Entertainment (movies, opera, concert, etc.) [_____]
 - e. Other (sleep, rest, talk, listen, shower, etc.) [_____]

9. Some activities you do when your device is not with you.
 - a. Sleep
 - b. Other (cook, watch TV, shower, etc.) [_____]

10. Give examples for each of the physical surroundings and environments mentioned below while your device is with you.
 - a. Indoors (home, office, buildings, etc.) [_____]
 - c. Public places (noisy – bus-stops, roads, malls, etc.) [_____]
 - d. Public places (silent – Concerts, Movies, etc.) [_____]
 - e. Vehicles (movement – car, bike, train, etc.) [_____]

11. How do you (would you) respond to incoming calls when you are busy in the following activities?
 - a. Middle of a conversation
 - b. Working
 - c. Not working (resting, passing time, etc.)
 - d. In a meeting (or movie theatre, concert, etc.)
 - e. Hands are busy (or eyes, ears, etc. for that matter)

12. The following are different ways a mobile device can interrupt a user:
 Active: ringer, music, voice, beeps, vibration.
 Passive: flash icons on screen, display message, LED, vibrate, voicemail, SMS.
 How would you prefer your device to interrupt you while performing the following activities?
 - a. In the middle of a call [_____]
 - b. In conversation [_____]
 - c. Sleeping [_____]
 - d. Trivial activities (Resting, passing time, etc.) [_____]
 - e. Work [_____]
 - f. Group (meeting, teaching, etc.) [_____]
 - g. Using the device (updates, notes, schedules, etc.) [_____]
 - h. Physical activities (exercise, walk, cook, etc.) [_____]
 - i. Travel (bus, car, train, plane, etc.) [_____]
 - j. Other [_____]

III. Assistant/Secretary Related Questions

1. Give examples of various activities you perform under each category
 - a. Visitor-related activities (such as appointments, scheduling, taking messages, etc.)
 - b. Phone-related activities (remainders, notifications, questions, etc.)
 - c. Computer-related activities (forms, emails, reports, letters, etc.)
 - d. Other activities
2. How many visitors (to the professor) do you receive/handle on average in a day?
 - a. 0 – 5
 - b. 6 – 10
 - c. 11 – 20
 - d. 20 and up
3. What kind of visitors come to meet with the professor? List the different sub-categories wherever necessary, and also mention the average number?
 - a. Students [_____]
 - b. Colleagues (other professors) [_____]
 - c. Dean (other V.I.P's) [_____]
 - d. Unknown visitors [_____]
 - e. Other [_____]
4. What actions do you perform when a visitor arrives at the office?
 - a. Obtain information about the visitor
 - b. Ask for her purpose of visit.
 - c. Ask if she has a scheduled appointment, or is visiting without.
 - d. Gather above information from other sources (diaries, calendars, time-tables, etc.)
 - e. Contact professor (directly, via phone, email, SMS, instant message, voicemail, etc.)
 - f. Deny permission to visit.
 - g. Allow visit without any questions.
 - h. Respond on behalf of the professor.
 - i. Request to be seated.
 - j. Check what the professor is doing.
 - k. Other [_____]
5. List above actions in the order in which you would perform them for the following categories (only list the corresponding letters against each action in sequence). You may omit any actions that you don't perform in a particular scenario. You may also list multiple actions by a '/' or 'or'. For example [a, b, c/d, e/j, f/g/h/i]
 - a. Based on appointment [Student: _____]

- [Colleague: _____]
 [V.I.P.: _____]
 [Unknown: _____]
- b. Based on casual drop-ins [Student: _____]
 [Colleague: _____]
 [V.I.P.: _____]
 [Unknown: _____]
- c. Based on purpose of visit [Student: _____]
 [Colleague: _____]
 [V.I.P.: _____]
 [Unknown: _____]
- d. Based on profs' activities [Student: _____]
 [Colleague: _____]
 [V.I.P.: _____]
 [Unknown: _____]

6. Indicate visitor priority as high or low based on the following categories.

- a. Based on appointment
 i. Student [____], Colleague [____], V.I.P. [____], Other [____]
- b. Casual drop-ins
 ii. Student [____], Colleague [____], V.I.P. [____], Other [____]
- c. Based on purpose of visit
 iii. Student [____], Colleague [____], V.I.P. [____], Other [____]
- d. Based on Professor's activities at the time of visit
 iv. Student [____], Colleague [____], V.I.P. [____], Other [____]

7. For the following categories what actions would you perform? Please select one or more actions from question 4, and list the letter corresponding to ones selected. You may omit any actions that you don't perform in a particular scenario. You may also list multiple actions by a '/' or 'or'. For example [a, b, c/d, e/j, f/g/h/i]

- a. High Priority & Prof is busy [____]
 a. High Priority & Prof is not busy [____]
 a. Low Priority & Prof is busy [____]
 a. Low Priority & Prof is not busy [____]

IV. PC-Usage & Instant Messaging Related Questions

1. On average, how many hours a day do you use your PC/Laptop? [____]
2. How much of this time is spent on work [____], and on entertainment [____]?
3. How much of the time on entertainment is spent on IM (chatting) [____]

4. List various software programs you use while working on your PC/Laptop?
5. List various software programs you use for Instant Messaging (IM, or chatting)?
6. List various software programs you use for entertainment? For example windows media player, powerdvd, any games, etc.
7. Select one or all of the following types of contacts in your instant message software.

<ol style="list-style-type: none"> a. Personal <ol style="list-style-type: none"> i. Family ii. Friends iii. Other [_____] 	<ol style="list-style-type: none"> b. Work <ol style="list-style-type: none"> i. Colleagues ii. Peers (work groups/project groups) iii. Supervisor/Boss/Prof.
---	--
8. Select one or all of the following activities do you perform with your IM software.
 - a. Send and receive messages.
 - b. Share work related documents.
 - c. Share music and pictures.
 - d. Other [_____]
9. While engaging in multiple conversations under the following scenarios, list the categories from question 7 in terms of priority. For example, if your family member, and colleague were online at the same time, and your colleague was discussing work, who would you communicate with first, and for majority of the time? List in sequence from high priority to low. For example, a list may look like [a(i), b(iii), b(ii), a(ii), b(i)].
 - a. Family, Peers, Friend all online and Peer is discussing work: [_____]
 - b. Family, Peers, Friend all online and all conversation is casual: [_____]
 - c. Friends, Supervisor are online and all conversation is casual [_____]
 - d. Family and Friends online and family is discussing something important. [_____]
 - e. Other possible scenarios you can think of
10. How many hours during your work time is the IM (chat) software running?
[_____]
11. Do you leave your task at hand to respond to IM messages? Or do you complete the task, and then respond?
[_____]
12. Does your answer to question 11 have any relation with who is messaging, or for what reason? For example, if you were expecting a message, you might leave an

important task at hand and respond to the message.

13. Do you consider a subject line an important addition to an IM message window? In other words, would a subject line (or purpose line) help you in deciding whether to respond to someone's message or not?
14. What actions do you perform immediately upon receiving a message, and immediately after responding to a message. You may select multiple actions.
 - a. Immediately upon receiving a message
 - i. Check whose message it is first.
 - ii. Directly switch to message window to respond.
 - iii. Save your task at hand and then switch.
 - c. Immediately after responding to message.
 - i. Go back to your task.
 - ii. Take a break (i.e. get a soda, or get up and stretch).
 - iii. Surf the web.
15. Select one or more of the following activities you do while sitting at your PC.
 - a. Work on documents and files.
 - b. Use the Internet (email, websites, or chat and instant messaging).
 - c. Listen to music.
 - d. Listen to music and work (or use the web).
 - e. Watch movies or play games.
 - f. Sit idle between tasks (gaze at screen, etc.)
 - g. Doze off.
 - h. Other [_____]
16. Select one or more from the following interruptions caused by your PC.
 - a. Pop-ups from websites
 - b. Virus alerts.
 - c. Updates and notifications about software.
 - d. Error messages.
 - e. Non-recoverable error message (PC crash, or program does not respond)
 - f. Scheduled tasks running automatically (virus scan, update installation, etc.)
 - g. Messages from other programs running in the background.
 - h. Other [_____]

APPENDIX B
CELL PHONE SURVEY 1

Examples of Notification, Notification Mode:

Ringer, Voice, Music, Beeps, Text Messages, Animations, Light (LED), Vibration

1. Circle all available notification options on your cell phone.

For Example: Beeps [Y/N]

- | | |
|-----------------|-------------------------------|
| a. Voice [Y/N] | e. Text messages [Y/N] |
| b. Beeps [Y/N] | f. Vibration [Y/N] |
| c. Music [Y/N] | g. Animations on screen [Y/N] |
| d. Ringer [Y/N] | h. LED (light) [Y/N] |

1a. What notification modes *would you prefer* in extremely high noise situations? Write down the TWO preferred ones from list in Question 1. [_____]

1b. What notification modes *would you prefer* in very quiet places and situations? Write down the TWO preferred ones from list in Question 1. [_____]

1c. Would you like to have a feature on your cell phone that indicates you of the caller's purpose (such as casual or emergency)? [Yes / No]

1d. Write down THREE preferred notification modes from the list in Question 1 for an EMERGENCY call from family, friends or work colleagues [_____]

2. Circle applicable cases listed below where you generally DO NOT pick up calls from WORK on your cell phone: { during/after dinner at home, late night out (after 9pm), formal dinners, campus parties, weekends at home (day, evening, night), with family at home }

3. Would you like to have a feature at certain locations such as inside theaters, class rooms, etc. that would enable your cell phone to automatically switch to a 'passive' mode such as text and vibrate, instead of the normal 'active' ringer and music.

[Yes / No]

4. If you were to forget your cell phone at the places listed below, circle ONE of the two actions you would like your cell phone to take.

- a. your home: {lock itself, do nothing} e. other peoples offices: {lock itself, do nothing}
 b. your office: {lock itself, do nothing} f. public transportation: {lock itself, do nothing}
 c. your vehicle: {lock itself, do nothing} g. public places: {lock itself, do nothing}
 d. others' homes: {lock itself, do nothing} h. other peoples vehicles: {lock itself, do nothing}

5. If you received an incoming call WHILE using your cell phone for activities A, B and C below, how would you prefer your phone to notify you? Write TWO modes for each from list below.

- 1) Ringer 2) vibrate 3) LED 4) music 5) beeps
 6) animations on screen 7) text message on screen 8) Voice

- A. Calling or receiving calls (with cell phone held to ear): []
 B. Listening to music, watching movies on cell phone (with earphones): []
 C. Using cell phone's calendar, notepad, or other screen based tools: []

6. Fill in the following table with TWO PREFERRED choices from the list below (WRITE NUMBERS ONLY)

- 1) Ringer 2) vibrate 3) LED 4) music 5) beeps
 6) Animations on screen 7) text message on screen 8) Voice

Assume phone is with you (or near you) and is kept switched on.

FOLLOW EXAMPLE	TYPES OF INCOMING CALLS				
Places and Activities	Emergency calls from Family/friend	Emergency calls from Work	Unknown Caller	Casual calls from Family/friend	Work Casual calls
EXAMPLE: In Classrooms	1, 5	1, 5	7, 3	3, 2	7, 2
1. At Meetings/Presentations					
2. In Classrooms					
3. Inside Cinema Theaters					
4. Fine Restaurants (Pine Club, etc)					
5. At Places of Worship					
6. In Elevators					
7. In Libraries					
8. In your car					

APPENDIX C CELL PHONE SURVEY 2

CELL PHONE QUESTIONNAIRE

List of Notification Modes: 1) Ringer 2) Voice 3) Music 4) Beeps 5) Text Messages 6) Animations 7) Light (LED) 8) Vibration

- From the above list, select the DEFAULT NOTIFICATION MODE on your cell phone?
(i.e. the mode your phone is generally on) []
- For each of the cases below, fill in Column 1 and Column 2 AS SHOWN IN EXAMPLES BELOW
 COLUMN 1: Do you switch from your default mode to a new notification mode? Choices are ALWAYS, SOMETIMES, or NEVER.
 COLUMN 2: If you wrote ALWAYS or SOMETIMES, what new modes do you use? Write down TWO modes from the list above.
 If you wrote NEVER, write down 'SAME AS DEFAULT'
 If you switch off your phone in certain cases, write down 'SWITCH OFF'

	COLUMN 1	COLUMN 2
EXAMPLE: Inside movie theater	Always	8, 5
EXAMPLE: Sleeping at night	Never	Same as Default
Home: Resting in the evening		
Sleeping at night		
Trivial activities (music, TV, books, surfing, etc.)		
Shower, bathroom		
Having dinner with family		
Cooking, cleaning, laundry		
Working on car, construction		
Mowing lawn, etc.		
Office work: Physical (eg. construction)		
Seated, PC based (eg. clerical)		
Work in noisy environment		
Travel based work		
In office (not working)		
In school outside of classrooms		
Vehicle: Driving vehicle yourself		
In vehicle as passenger		
On an airplane		

Other: Inside movie theatre (during film)	
Outside movie theater	
Department stores and malls	
Walking on streets, in parks, etc.	
Restaurants (TGIF, Wendys,...)	
Friends/relatives place (informal)	
Formal dinners and parties	
Noisy public places (subway etc.)	
Sports: Outdoor (running, biking, etc.)	
Indoor (gym, etc.)	
General family time (dinners, chat, etc.)	
In Meetings/Presentations	
In Classrooms	
In Inside Cinema Theaters	
In Places of Worship	
In Elevators	
In Hospitals	
In Libraries	

APPENDIX D
CELL PHONE SURVEY 3

CELL PHONE QUESTIONNAIRE

FOR EACH OF THE ACTIVITIES OR PLACES IN THE TABLE:

COLUMN 1: Where is your phone generally located? Choices are

- Pocket Belt
Hand Hand
Purse Nearby
Not with you Other (specify)

COLUMN 2: How would you like your cell phone to notify you in case of an incoming call?

Select THREE modes from below

1. Ringer
2. Voice
3. Music
4. Beeps
5. Text Messages
6. Animations
7. Light (LED)
8. Vibration

COLUMN 3: From the list below, how do you decide whether to answer the phone or not?
Rank from highest priority to lowest.

- a. Based on Caller
 - b. Based on Purpose
 - c. Based on Time
 - d. Based on the Situation
- If none, write Doesn't matter

ACTIVITIES AND PLACES	COLUMN 1	COLUMN 2	COLUMN 3
Example: driving vehicle yourself	Nearby	1, 2, 4	d, b, a, c
Home: Resting in the evening			
Sleeping at night			
Trivial activities (music, TV, books, surfing, etc.)			
Shower, bathroom			
Having dinner with family			
Cooking, cleaning, laundry			
Office work: Physical (eg. construction)			
Seated, PC based (eg. clerical)			
Work in noisy environment			
Travel based work			
In office (not working)			
In school outside of classrooms			
Vehicle: Driving vehicle yourself			
In vehicle as passenger			
Other: Inside movie theatre (during film)			
Outside movie theater			
Department stores and malls			
Walking on streets, in parks, etc.			
Restaurants (TGIF, Wendys...)			
Friends/relatives place (informal)			
Formal dinners and parties			
Noisy public places (subway etc.)			
Sports: Outdoor (running, biking, etc.)			
Indoor (gym, etc.)			

APPENDIX E

SAMPLE RULESETS FOR CELL PHONE SCENARIO

The attributes and attribute values for the cell phone scenario are listed below. These along with information elicited from user surveys and field studies are used to develop generic rules and policies to build training datasets for the interruption-aware decision tree models.

```
@attribute decibel {lessthan20db, 20to55db, 55to85db, morethan85db}
@attribute soundtype {speech, noise, music}
@attribute motion {high, medium, low, none}
@attribute time {day, evening, night}
@attribute usage {call, music, screen, none}
@attribute light {bright, medium, dark}
@attribute day {weekday, weekend}
@attribute caller {family, friend, work, unknown, none}
@attribute context {casual, intended, emergency, unknown}
@attribute location {office, home, vehicle, other}
@attribute proximity {lessthan3ft, 3to20ft, morethan20ft}
@attribute sensitivity {yes, no, notapplicable}
```

```
@attribute response {ringer, voice, beep, LED, lockphone, donothing,
vibrate, text}
```

A few sample rulesets that have been developed are shown below

VI. Phone with(on) user, and phone in active usage

1. sensitivity: yes

1a. context: emergency
usage: call → voice
music → vibrate
screen → LED

1b. context: intended
usage: call → beep
music → vibrate
screen → LED

2d. context: unknown → LED

II. Phone near user, phone not in use, and user not in sensitive area

1. caller: family, friend, work

1a. context: casual
 location: office
 motion: high, medium
 decibel: morethan85db → beep
 55to85db → beep
 20to55db → LED
 lessthan20db → LED
 motion: low, none
 decibel: morethan85db → beep
 55to85db
 soundtype: noise → LED
 speech → LED
 music → beep

6. caller: friend, work

6a. context: casual
 location: home
 time: night
 decibel: lessthan20db → LED
 morethan85db → ringer

VII. Phone with(on) user, phone not in use, user in sensitive area

1. context: emergency
 caller: family, friend
 location: home, vehicle → voice
 office, other
 decibel: morethan85db
 soundtype: noise → voice

4. context: unknown
 caller: unknown → LED

X. phone not with, or near user and there is an incoming call

1. context: emergency
 sensitivity: yes → LED
 no, notapplicable
 location: home, vehicle, other → voice
 office → LED

2. context: casual
 sensitivity: yes → text
 no, not applicable

2a. location: home
 caller: family, friend
 decibel: morethan85db → ringer
 55to85db → ringer

APPENDIX F

SAMPLE RULESETS FOR VIRTUAL SECRETARY SCENARIO

The attributes and attribute values for the virtual secretary scenario are listed below. These along with information elicited from user surveys and field studies are used to develop generic rules and policies to build training datasets for the interruption-aware decision tree models.

```
@attribute onphone {yes, no}
@attribute motion {yes, no}
@attribute chairpressure {yes, no}
@attribute soundtype {speech, music, noise, none}
@attribute presence {yes, no}
@attribute busyattribute {set, reset}
@attribute calendar {away, in}
@attribute newvisitor {student, colleague, vip, other, none}
@attribute purpose {appointment, dropin, emergency, none}
@attribute currentvisitor {student, colleague, vip, other, none}
@attribute busytime {zero, 10min, 11to30min, morethan30min }
@attribute awaytime {zero, 10min, 11to30min, morethan30min}

@attribute allowvisit {allow, checklater, wait, takemssg, donothing}
```

A few sample rulesets are shown below

No new visitor

1. newvisitor: none → donothing

presence: no

```
1. presence: no
    calendar: away
        newvisitor: student, colleague, vip, other → takemssg

    calendar: in
        awaytime: 10min
        purpose: appointment → wait
        emergency → wait
        dropin
            newvisitor: student → wait
            colleague → takemssg
            vip → takemssg
```

awaytime: 11to30min
purpose: appointment
newvisitor: student → wait
colleague → takemssg

emergency → takemssg
dropin → checklater

awaytime: morethan30min
newvisitor: colleague, vip, other → takemssg

presence: yes

on phone

1. presence: yes
onphone: yes
busytime: 10min
newvisitor: student, other
purpose: appointment → wait
emergency → allow

colleague, vip → allow

busytime: 11to30min
newvisitor: student, other
purpose: appointment → checklater
emergency → allow

colleague, vip → allow

busytime: morethan30min
newvisitor: student, other
purpose: dropin → checklater
emergency → allow

colleague, vip → allow

not on phone and not busy

1. presence: yes
onphone: no
busyattribute: reset
currentvisitor: none
newvisitor: student, colleague, vip, other → allow

not on phone and busy (but no previous visitor inside)

1. presence: yes
onphone: no
busyattribute: set
currentvisitor: none
newvisitor: student, other
purpose: appointment → wait

BIBLIOGRAPHY

Abowd, G.D., C.G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A mobile context-aware tour guide," *Wireless Networks*, Vol. 3, No. 5, pp. 421-433, October, 1997.

Adamczyk, P.D., and B.P. Bailey, "If Not Now When? The Effects of Interruptions at Various Moments within Task Execution," In *CHI*, pp. 271-278, 2004.

Alreck, P.L., R.B. Settle, *THE SURVEY RESEARCH HANDBOOK: GUIDELINES AND STRATEGIES FOR CONDUCTING A SURVEY*, 2nd Ed., McGraw Hill, 1995.

Altmann, E.M., and J.G. Trafton, "Task interruption: Resumption lag and the role of cues," In *Proceedings of the 26th annual conference of the Cognitive Science Society*, Erlbaum Associates, pp. 42-47, 2004.

Arroyo, E., T. Selker, and A. Stouffs, "Interruptions as multimodal outputs: Which are the less disruptive?," In *ICMI'02*, pp. 479-483, 2002.

Asthana, A., M. Cravatts, and P. Krzyzanowski, "An indoor wireless system for personalized shopping assistance," In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, pp. 69-74, December, 1994.

Avrahami D., and S.E. Hudson, "Balancing performance and responsiveness using an augmented instant messaging client," In *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW'04)*, New York: ACM Press, 2004.

Bailey, B.P., J.A. Konstan, and J.V. Carlis, "The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface," In *Proceedings of INTERACT 2001*, NCP, Tokyo, Japan, 2001.

Bailey, B.P., J.A. Konstan, and J.V. Carlis, "Adjusting Windows: Balancing Information Awareness with Intrusion," In *Proceedings of HFWeb*, 2000.

Bainbridge, L., "Diagnostic skills in process operation," In *Proceedings of the 1984 International Conference on Occupational Ergonomics*, Vol. 2, pp. 1-10, 1984.

Bannon L., A. Cypher, S. Greenspan, and M.L. Monty, "Evaluation and analysis of users' activity organization," In CHI'83, pp. 54-57, 1983.

Barkhuus, L., A.K. Dey, "Location-based services for mobile telephony: a study of users' privacy concerns," INTERACT 2003: 9th IFIP TC13 International Conference on Human-Computer Interaction, September 1-5, 2003.

Bautsch, H., Granger, J., Karnjate, T., Kahn, F., Leveston, Z., and Niehus, G., "An investigation of mobile phone use: A socio-technical approach," Socio-technical Systems in Industry, Department of Industrial Engineering, University of Wisconsin-Madison, 2001.

Beadle, H.W.P., B. Harper, G.Q. Maguire Jr., and J. Judge, "Location aware mobile Computing," In Proceedings of IEEE/IEE International Conference on Telecommunications, Melbourne, Australia, April, 1997.

Bederson, B.B., "Audio augmented reality: A prototype automated tour guide," In Proceedings of Conference on Human Factors and Computing Systems, CHI '95, Denver, CO, pp. 210-211, 1995.

Behforooz, A., and F.J. Hudson, SOFTWARE ENGINEERING FUNDAMENTALS, Oxford University Press, 1996.

Beigl, M. "MemoClip: A location-based remembrance appliance," Personal Technologies, Vol. 4, No. 4, pp. 230-233, September, 2000.

Bolt, R.A., "Put-that-there: Voice and gesture at the graphics interface," Computer Graphics, Vol. 14, No. 3, pp. 262-270, July, 1980.

Boutilier, C., R.I. Brafman, C. Domshlak, H.H. Hoos, and D. Poole, "CP-nets: A tool for representing and reasoning with conditional Ceteris Paribus preference statements," In Journal of Artificial Intelligence Research, Vol. 21, pp. 135-191, 2004.

Bradner, E., Kellogg, W. A. and Erickson, T., "Babble: Supporting Conversation in the Workplace," SIGGROUP Bulletin, ACM Press, Vol. 19, No. 3, December, pp 8-9, 1998.

Brandimonte, M., G.O., Einstein, and M.A. McDaniel, PROSPECTIVE MEMORY: THEORY AND APPLICATIONS. Erlbaum Associates, Mahwah, NJ, 1993.

Brewster, S.A., P.C. Wright, and A.D.N. and Edwards, "The Design and Evaluation of an Auditory Enhanced Scrollbar," In CHI '94, pp. 173-179, 1994.

Brusilovsky, P., E. Schwarz, and G. Weber, "ELM-ART: An intelligent tutoring system on World Wide Web," In Proceedings of Third International Conference on Intelligent Tutoring Systems, Montreal, June 12-14, pp. 261-269, 1996..

Campbell, R., A. Ravi, and A. Mahajan, "ConChat: A Context-Aware Chat Program", IEEE Pervasive Computing, pp. 52-58, September, 2002.

Capuano, N., M. Marsella, S. Salerno, "ABITS: an agent-based intelligent tutoring system for distance learning," In Proceedings of the International Workshop on Adaptive and Intelligent Web-based Educational Systems, International Conference on Intelligent Tutoring Systems (ITS 2000), Montreal, Canada, 2000.

Castells, M., M.F. Ardevol, J.L. Qiu, and A. Sey, "The Mobile Communication Society: A Cross Cultural Analysis of Available Evidence on the Social Uses of Wireless Communication Technology," Technical Report for the International Workshop on Wireless Communication Policies and Prospects, University of Southern California, Los Angeles, CA, 2004.

Chen, G., M. Li, and D. Kotz, "Design and implementation of a large-scale context fusion network," In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), 2004.

Cohen, S., "Aftereffects of stress on human performance and social behavior: A review of research and theory," Psychological Bulletin, Vol. 88, No. 1, pp. 82-108, 1980.

Cohen, P.R., M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, and J. Clow, "QuickSet: Multimodal Interaction for Distributed Applications", Proceedings of the Fifth ACM International Multimedia Conference, New York, NY, pp. 31-40, 1997.

Conner, C.T., B.L. Feldman, E.B. Moreau, K. Lebo, and C. Kaschub, "A practical guide to experience-sampling procedures," Journal of Happiness Studies, Vol. 4, pp. 53-78, 2001.

Cooper, A., THE INMATES ARE RUNNING THE ASYLUM: WHY HIGH-TECH PRODUCTS DRIVE US CRAZY AND HOW TO RESTORE THE SANITY, Sams, 2nd Edition, 2004.

Czerwinski, M., E. Cutrell, and E. Horvitz, "Instant Messaging: Effects of Relevance and Time," In People and Computers XIV: Proceedings of HCI 2000, Vol. 2, pp. 71-76, 2000.

Czerwinski, M., E. Cutrell, and E. Horvitz, "Instant messaging and interruption: Influence of task type on performance," In OZCHI 2000 Conference Proceedings, pp. 356-361, 2000.

Dabbish L.A., and R.S. Baker, "Administrative assistants as interruption mediators," In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'03), New York, pp. 1020-1021, 2003

Dekel, U., and S. Ross, "Eclipse as a Platform for Research on Interruption Management in Software Development," Eclipse Technology Exchange Workshop at OOPSLA'04, Vancouver, BC, Canada, October, 2004.

Dertouzos, M., THE UNFINISHED REVOLUTION: Human Centered Computers and what they can do for us, Harper Collins, New York, NY, 2001.

Detweiler, M.C., S.M. Hess, and M.P. Phelps, "Interruptions and working memory," Technical report, The Pennsylvania State University, Department of Psychology.

Dey, A. K. and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI 99), May 15 – 20, 1999, pp. 434 – 441.

Dey, A.K., G.D. Abowd, "CyberMinder: A Context-Aware System for Supporting Reminders," Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing, Vol. 1, pp. 172, 2000.

Dey, A.K., "Understanding and Using Context," Personal and Ubiquitous Computing Journal, Vol. 5, No. 1, pp. 4-7, 2001.

Domshlak, C., R. Brafman, S.E. Shimony, "Preference-based configuration of web page content," In Proceedings of Seventeenth International Joint Conference on Artificial Intelligence, Seattle, pp. 1451-1456, 2001.

Erickson, T., "Some problems with the notion of context-aware computing," In the Journal of Communications of the ACM, Vol. 45, No. 2, pp. 102-104, 2002.

Ertas, A., and J.C. Jones, THE ENGINEERING DESIGN PROCESS, John Wiley and Sons Inc., 1992.

Flanagan, J., I. Marsic, A. Medl, G. Burdea, and J. Wilder, "Multimodal Human/Machine Communication," Multimedia Communications, 10th Tyrrhenian International Workshop on Digital Communications, Sept. 16-18, Ischia, Italy, 1998.

Franke, J., J. Daniels, and D. McFarlane, "Recovering Context After Interruption," 24th Annual Meeting of the Cognitive Science Society, Fairfax, VA, August 8-10, 2002.

Gardner Jr., P.C., "A System for the Automated Office Environment," In IBM Systems Journal, Vol. 20, No. 3, pp. 321-345, 1981.

Gibbs W.W., "Considerate computing," In Scientific American, Vol. 292, No. 1, pp. 54-61, 2005.

Gievska S., and J. Sibert, "Using task context variables for selecting the best timing for interrupting users," In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies*, New York: ACM Press, pp. 171-176, 2005.

Gillie, T., and D.E. Broadbent, "What Make Interruptions Disruptive? A Study of Length, Similarity, and Complexity," *Psychological Research*, Vol. 50, pp. 243-250, 1989.

Goschke, T., J. Kuhl, "Representation of intentions: Persisting activation in memory," In *Journal of Experimental Psychology: Learning, Memory, & Cognition*, Vol. 19, No. 5, pp. 1211-1226, 1993.

Gruen, D., C. Sidner, C. Boettner, and C. Rich, "A collaborative assistant for email," In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA, May, 1999.

Ha, V., and P. Haddawy, "Toward case-based preference elicitation: Similarity measures on preference structures," In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI, pp. 193-201, 1998.

Halverson, C., L.E. Julia, A.A. Cheyer, "Walking on Water: The task of supporting-not coming between-people and work," *COOP'2000*, Sophia-Antipolis, France, 2000.

Hansson, R., P. Ljungstrand, and J. Redstrom, "Subtle and public notification cues for mobile devices," *Lecture Notes in Computer Science, Ubicomp*, Vol. 22, No. 1, pp. 240-247, 2001.

Harrison, B.L., H. Ishii, K.J. Vincente, and W.A.S. Buxton, "Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention," In *Conference Proceedings on Human Factors in Computing Systems (CHI '95)*, pp. 317-324, 1995.

Hayes, P.J., "The Frame Problem and Related Problems in Artificial Intelligence," *Technical Report CS-TR-71-242*, Stanford University Press, 1971.

Horvitz, E., C. Kadie, T. Paek, and D. Hovel, "Models of Attention in Computing and Communication: From Principles to Applications," *Communications of the ACM*, Vol. 46, No. 3, pp. 52-59, 2003.

Horvitz, E., and J. Apacible, "Learning and Reasoning about Interruption," *Proceedings of the ACM International Conference on Multimodal Interfaces*, pp. 20-27, 2003.

Horvitz E., A. Jacobs, and D. Hovel, "Attention sensitive alerting," In *Proceedings of the 15th Conference in Conference on Uncertainty and Artificial Intelligence (UAI'99)*, San Francisco: Morgan Kaufmann, pp. 305-313, 1999.

Howard, R.A., and J.E. Matheson, READINGS ON THE PRINCIPLES AND APPLICATIONS OF DECISION ANALYSIS, Strategic Decision Group, Menlo Park, CA, 1984.

Hudson S., J. Fogarty, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang, "Predicting human interruptability with sensors: A Wizard of Oz feasibility study," In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI'03), New York: ACM Press, pp. 257-264, 2003.

Intille, S.S., J. Rondoni, C. Kukla, I. Anaconda, and L. Bao, "A context-aware experience sampling tool," In Proceedings of the Conference on Human Factors in Computing Systems: Extended Abstracts, New York, NY, pp. 972-973, 2003.

Ishii, H., B. Ullmer, "Tangible Bits: Towards seamless interfaces between people, bits and atoms," CHI 97 Conference on Human Factors in Computing systems, Atlanta, Georgia, March, 1997.

Keeney, R.L., and H. Raiffa, DECISIONS WITH MULTIPLE OBJECTIVES: PREFERENCES AND VALUE TRADEOFFS, Wiley, New York, 1976.

Khalil, A., and K. Connelly, "Context-aware Configuration: A study on improving cell phone awareness," In Proceedings of Context'05, Paris, France, July, 2005.

Khalil, A., and K. Connelly, "Improving Cell Phone Awareness by Using Calendar Information," In Proceedings of INTERACT'05, Rome, Italy, 2005.

Kortuem, G., M. Bauer, T. Heiber, and Z. Segall, "Netman: The design of a collaborative wearable computer system," In ACM Journal on Mobile Networks and Applications (MONET), Vol. 4, No. 1, 1999.

Lahaie, S., and D. Parkes, "Applying learning algorithms to preference elicitation," In Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), New York, NY, 2004.

Larson, R., and M. Csikszentmihalyi, "The experience sampling method," New Directions for Methodology of Social and Behavioral Science, Vol. 15, 1983, pp. 41-56.

Latorella, K.A., "Investigating Interruptions: Implications for flight deck performance," NASA Technical Memorandum 209707, National Aviation & Space Administration, Washington, DC, 1999.

Lee, J.C., "Spatial User Interfaces: Augmenting Human Sensibilities in a Domestic Kitchen," Master Thesis, School of Architecture and Planning, MIT Media Laboratory, 2005.

Ling, R., and L. Haddon, "Mobile telephony, mobility, and the coordination of everyday life," In *Machines That Become Us: The Social Context of Personal Communication Technology Conference*, New Brunswick, pp. 245-265, 2003.

Liu, K.K. "A Personal, Mobile System for Understanding Stress and Interruptions," S.M. Thesis, Massachusetts Institute of Technology, 2004.

Lockerd, A., H. Pham, T. Sharon, and T. Selker, "Mr.Web: An Automated Interactive Webmaster," In *Computer Human Interaction'03*, Vol. 4, 2003.

Lonergan, D., "Analyst Watch," Lucent Technologies Report, pp. 30-31, 2005

Madhavan, D., and K. Funk, "Cockpit task management errors in critical in-flight incidents," In *Seventh International Symposium on Aviation Psychology*, pp. 970-974, 1993.

Maes, P., "Agents that Reduce Work and Information Overload," In *CACM*, Vol. 37, No. 7, July, 1994.

Mark, G., V.M. Gonzalez, J. Harris, "No Task Left Behind? Examining the Nature of Fragmented Work," In *CHI: Interruptions and Attention 1*, April, Portland, Oregon, pp. 321-330, 2005.

Marmasse, N., and C. Schmandt, "Location-aware information delivery with ComMotion," In *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*, pp. 157-171, Bristol, UK, September, 2000.

Mc. Gee, D.R., and P. Cohen, "Exploring Handheld, Agent-based, Multimodal Collaboration," In *Proceedings of the Workshop on Handheld Collaboration at the Conference on Computer Supported Cooperative Work*, Seattle, WA, 1998.

McCrickard, D.S., C.M. Chewar, "Attuning notification design to user goals and attention costs," In *Communications of the ACM*, Vol. 46, No. 3, pp. 67-72, 2003.

McCrickard, D.S., R. Catrambone, and J.T. Stasko, "Evaluating Animation in the Periphery as a Mechanism for Maintaining Awareness," *Proceedings of INTERACT 2001*, Tokyo, Japan, July, 2001, pp. 148-156.

McDermott, D., "Artificial intelligence meets natural stupidity. *Mind Design: Philosophy, Psychology, Artificial Intelligence*," MIT Press, Cambridge, MA, pp. 143-160, 1981.

McFarlane, D.A., "Coordinating the Interruption of People in Human-Computer Interaction," In *Proceedings of Interact'99*, IOS Press, 1999, pp. 295-303.

McFarlane, D.C., "Interruption of people in human-computer interaction: A general unifying definition of human interruption and taxonomy," NRL Formal Report, NRL/FR/5510—97-9870, Washington, DC, Naval Research Laboratory, 1997.

McFarlane, D.C., and Latorella, K. A., "The scope and importance of human interruption in HCI design," *Human-Computer Interaction*, Vol. 17, pp. 1-61, 2002.

McFarlane, D.C., "Comparison of Four Primary Methods for Coordinating the Interruption of People in Human-Computer Interaction," In *HCI'02*, Vol. 17, pp. 63-139, 2002.

Meyer, S., and A. Rakotonirainy, "A survey of research on context-aware homes," *Conference on Research and Practice in Information Technology Series, Proceedings of the Australasian information security workshop conference on ACSW frontiers*, Vol. 21, Adelaide, Australia, pp 159-168, 2003.

Middendorf, W.H., *DESIGN OF DEVICES AND SYSTEMS*, 2nd Edition, Marcel Dekker Inc., 1989.

Milewshi, A.E., and T.M. Smith, "Providing Presence Cues to Telephone Users," In *Proceedings of CSCW 2000*, Philadelphia, PA, 2000.

Mitchell, T., *MACHINE LEARNING*, McGraw-Hill Inc., New York, NY, 1997.

Mitchell, T., R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience With a Learning Personal Assistant," *Communications of the ACM*, Vol. 37, No. 7, July, pp. 81-91, 1994.

Monk, C.A., D.A. Bohem-Davis, and J.G. Trafton, "The attentional costs of interrupting task performance at various stages," In *Proceedings of 46th Annual Meeting of the Human Factors and Ergonomics Society (HFES 2002)*, Santa Monica, pp. 1824-1828, 2002.

Muller, M., S. Kuhn, "Participatory Design," In *Communications of the ACM*, Vol. 36, No. 6, pp. 24-38, 1993.

Norman, D.A., "COGNITIVE ENGINEERING: USER CENTERED SYSTEM DESIGN, D.A. Norman and S.W. Draper, Editors, Lawrence Erlbaum, Hillsdale, NJ, 1986.

Oppenheim, A.N., *QUESTIONNAIRE DESIGN, INTERVIEWING AND ATTITUDE MEASUREMENT*, Pinter Publishers, New York, 1992.

Oviatt, S., A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction," *Proceedings of Conference on Human Factors in Computing Systems: CHI '97*, ACM Press, New York, NY, 1997, pp. 415-422.

Palen, L., M. Salzman, and E. Youngs, "Going Wireless: Behavior and Practice of New Mobile Phone Users," In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2000)*, ACM Press, Philadelphia and New York, pp. 201-210, 2000.

Patalano, A.L., C.M. Seifert, "Opportunistic planning: Being reminded of pending goals," In *Cognitive Psychology*, Vol. 34, pp. 1-36, 1997.

Pedersen, E.R., "Calls.calm: Enabling Caller and Callee to Collaborate," In *Proceedings of CHI 2001*, Seattle, USA, 2001.

Perguini, S., and M.A., Gonzalves, "Recommendation and personalization: a survey," Technical Report CS.IR/0205059, Computing Research Repository: 2002.

Philipose, M., K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Fox, H. Kautz, and D. Hähnel, "Inferring Activities from Interactions with Objects," *IEEE Pervasive Computing*, October, pp. 50-57, 2004.

Postema, L.D., "Automated Highway Systems: Incremental Deployment as a Solution for the Future of Transportation," *IEEE Washington Internships for Students of Engineering (WISE) Report*, Washington, DC, 1998.

Quinlan, J.R., "Induction of Decision Trees," *Machine Learning*, Vol. 1, No. 1, 1986, pp. 81-106.

Quinlan, J.R., *C4.5: PROGRAMS FOR MACHINE LEARNING*, Morgan Kaufmann Publishers, San Mateo, California, 1993.

Raisamo, R., "A multimodal user interface for public information kiosks," *Proceedings of 1998 Workshop on Perceptual User Interfaces*, San Francisco, CA, 1998.

Ramchurn, S.D., B. Deitch, M.K. Thompson, D.C. DeRoure, N.R. Jennings, and M. Luck, "Minimising intrusiveness in pervasive computing environments using multi-agent negotiation," In *Proceedings of the first International Conference on Mobile and Ubiquitous Systems*, Boston, 2004, pp. 364-372.

Rauschert, I., P. Agrawal, S. Fuhrmann, I. Brewer, H. Wang, R. Sharma, G. Cai, and A. MacEachren, "Designing a human-centered, multimodal GIS interface to support emergency management," In the *10th ACM Symposium on Advances in Geographic Information Systems*, Washington, DC, USA, 2002.

Rillings, J.H. "Automated Highways", Scientific American, October, 1997.

Rouncefield, M., J.A. Hughes, T. Rodden, and S. Viller, "Working with constant interruption: CSCW and the small office," Proceedings of the CSCW '94 Conference on Computer-Supported Cooperative Work, New York, NY, 1994, pp. 275-286.

Ryszard S.M., I. Bratko, and M. Kubat, MACHINE LEARNING AND DATA MINING: METHODS AND APPLICATIONS, John Wiley & Sons Ltd, 1998.

Savioja, P., 2004, "Alarms – Necessary Interruptions?," T.121-900 Seminar on User Interfaces and Usability, HUT, SoberIT, 2004.

Schmidt, A., K.A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde, "Advanced interaction in context," In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, Karlsruhe, Germany, pp. 89-101, September, 1999.

Sharon, T., H. Lieberman, and T. Selker, "A Zero-input Interface for Leveraging Group Experience in Web Browsing," In Intelligent User Interfaces'03, Vol. 1, 2003.

Shneiderman, B., DESIGNING THE USER INTERFACE: STRATEGIES FOR EFFECTIVE HUMAN-COMPUTER INTERACTION, Third Edition, Addison-Wesley, Reading, MA, 1997.

Siewiorek D., A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, and K. Reiger, "SenSay: a context-aware mobile phone," In Proceedings of the IEEE international symposium on wearable computers, White Plains, New York, October, 2003.

Simon, H., "Invariants of Human Behavior," Annual Reviews of Psychology, Vol. 41, pp. 1-20, 1990.

Singh, P., and W. Williams, "LifeNet: a propositional model of ordinary human activity," In Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP), 2003.

Speier, C., I. Vessey, and J.S. Valacich, "The Effects of Interruptions, Task Complexity, and Information Presentation on Computer-Supported Decision-Making Performance," In Decision Sciences, Vol. 34, pp. 771, 2003.

Stafford-Fraser, Q., and P. Robinson, "BrightBoard: A Video-Augmented Environment," Conference on Human Factors in Computing Systems, April, pp. 134-141, 1996.

Stiemerling, O., H. Kahler, and V. Wulf, "How to Make Software Softer - Designing Tailorable Applications," In Proceedings of DIS'97, August, Amsterdam, Holland, 1997.

- Swere, E., D. Mulvaney, and I. Sillitoe, "Real-time Incremental Decision Tree Generation for Embedded Applications," In 2nd ESC Division Mini-Conference, Leicestershire, UK, January, pp. 6-8, 2005.
- Trafton, J. G., E.M. Altmann, D.P. Brock, and F.E., "Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal," In *International Journal of Human-Computer Studies*, Vol. 58, pp. 583-603, 2003.
- Utgoff, P.E., "ID5: An incremental ID3," In *Proceedings of the Fifth International Conference on Machine Learning*, pp. 107-120, 1988.
- Van Laerhoven, K., "Online Adaptive Context Awareness, starting with low-level sensors," Masters Thesis at the Free University of Brussels, Brussels, Belgium, 1999.
- Van Laerhoven, K., A. Schmidt, and H.W. Gellersen, "Multi-Sensor Context-Aware Clothing," In *Proceedings of the Sixth International Symposium on Wearable Computers, ISWC 2002*, Seattle, WA, 2002.
- Van Solingen, R., E. Berghout, F. Van Latum, "Interrupts: Just A Minute Never Is," In *IEEE Software*, Vol.15, No. 5, pp. 97- 103, 1998.
- Want, R., A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge location system," *ACM Transactions on Information Systems*, Vol. 10, No. 1, pp. 91-102, January, 1992.
- Welch, R.B., and D.H. Warren, "Intersensory Interactions," In *Handbook of Perception and Human Performance*, Chapter 25, New York: Wiley, 1986.
- White, J.R., and T.L. Booth, "Towards an Engineering Approach to Software Design," In *Proceedings of the 2nd International Conference on Software Engineering*, San Francisco, California, October, pp. 214-222, 1976.
- Witten, I.H., and F. Eibe, *DATA MINING: PRACTICAL MACHINE LEARNING TOOLS AND TECHNIQUES*, Morgan Kaufmann Publications, 2nd Edition, 2005.
- Xiaohang, W., "The Context Gateway: A Pervasive Computing Infrastructure for Context Aware Services," Report submitted to School of Computing, National University of Singapore & Context -Aware Department., Institute for Infocomm Research, November, 2003.
- Yan H., and T. Selker, "A Context-Aware Office Assistant," *ACM International Conference on Intelligent User Interfaces (IUI-2000)*, New Orleans, LA, January, 2000.
- Yanagida, Y., S. Kawato, H. Noma, N. Tetsutani, and A. Tomono, "A Nose-tracked, Personal Olfactory Display," In *Proceedings of the SIGGRAPH 2003 conference on Sketches & applications*, San Diego, CA, 2003.

Yankelovich, N., and C.D. McLain, "Office Monitor," CHI '96 Conference on Human Factors in Computing Systems, Vancouver, British Columbia, Canada, April 14-18, 1996.

WEBSITES

AURA, Distraction-free Ubiquitous Computing Project in Carnegie Mellon University, Available at: <http://www-2.cs.cmu.edu/~aura/>, Last accessed 22, March, 2005.

BlueEyes, IBM Website, <http://www.almaden.ibm.com/cs/BlueEyes/>, Last accessed 16, March, 2005.

BodyMedia, <http://www.bodymedia.com/>, Last accessed 20, March, 2006.

ENDEAVOUR, The Endeavour Expedition: Charting the Fluid Information Utility, Available at: <http://endeavour.cs.berkeley.edu/>, Last accessed 22, March, 2005.

EPA Levels Document, "Protective Noise Levels: Condensed Version of the EPA Levels Document," <http://www.nonoise.org/library/levels/levels.htm>, Last Accessed 2, April, 2006.

FitSense, <http://www.fitsense.com/>, Last accessed 20, March, 2006.

Kellogg, W.A., Erickson, T., Halverson, C., Laff, M., Malkin, P., Wolf, T., "Loops: A Blended Synchrony, Socially Translucent Conversation Environment for the Web," Available at: <http://www.research.ibm.com/SocialComputing/Papers/posterDemo3b.pdf>, 2001.

MIT Media Lab, <http://affect.media.mit.edu/>, Last accessed 26, February, 2005.

Nokia, "Developer Resources," Available at: <http://www.nokia.com/A402541>, Last Accessed 5, April, 2006.

Oz Project, <http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/oz.html>, Last accessed 24, March, 2005.

Palm, "Palm OS Tools," Available at: <http://www.palmos.com/dev/tools/>, Last Accessed 5, April, 2006.

Polar, <http://www.polarusa.com/>, Last accessed 20, March, 2006.

PORTOLANO, An Expedition into Invisible Computing at the University of Washington, Available at: <http://portolano.cs.washington.edu/>, Last accessed 22, March, 2005.

R702033606