

**Data Mining-Based Inhabitant Action Predictor for Smart Homes using
Controlled Synthetic Data**

by

Varadharajan Sridhar Pundi

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Master of Science

Department of Computer Science
Faculty of Graduate Studies

University of Manitoba

February 2008

Copyright ©2008 by Varadharajan Sridhar Pundi

THE UNIVERSITY OF MANITOBA
FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION

**Data Mining-Based Inhabitant Action Predictor for Smart Homes Using Controlled
Synthetic Data**

BY

Varadharajan Sridhar Pundi

**A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of
Manitoba in partial fulfillment of the requirement of the degree**

MASTER OF SCIENCE

Varadharajan Sridhar Pundi © 2008

**Permission has been granted to the University of Manitoba Libraries to lend a copy of this
thesis/practicum, to Library and Archives Canada (LAC) to lend a copy of this thesis/practicum,
and to LAC's agent (UMI/ProQuest) to microfilm, sell copies and to publish an abstract of this
thesis/practicum.**

**This reproduction or copy of this thesis has been made available by authority of the copyright
owner solely for the purpose of private study and research, and may only be reproduced and copied
as permitted by copyright laws or with express written authorization from the copyright owner.**

DEDICATION

I dedicate my master's degree to my father Sridhar Sampath Pundi and my mother Padma-
vathi Sridhar.

ACKNOWLEDGMENTS

I am grateful to my supervisor Dr. Neil Arnason for the guidance I received from him through-out my masters program right from the day go and it is under his guidance most of the ideas were conceptualized and refined in my research. I am grateful to my co-supervisor Dr. Rasit Eskicioglu who helped me in choosing the right courses and the research area. I thank Dr. Peter Graham for agreeing to be my internal examiner and for all those useful brainstorming sessions we had on my research work. I thank Dr. Jeff Diamond for agreeing to be my external examiner. Finally, I sincerely thank the Province of Manitoba, for having funded me with a two year Manitoba Graduate Scholarships that enabled me to pursue my Masters research.

ABSTRACT

Smart home research has led to the development of many sophisticated network protocols, smart appliances, and home gateway technologies. A smart home is a networked home containing various electrical and electronic devices controlled by a home gateway which manages the appliances and connects the home to service providers via the Internet. Smart homes are generally designed to assist people with cognitive impairments, seniors, and/or people with physical disabilities in their day-to-day activities. A key element in building such a user-adaptive smart home is to fully utilize the computational capabilities and automate the working of the smart appliances based on the inhabitants' appliance usage patterns.

The goal of this thesis is to build a system which will assist device automation in smart homes based on the device usage patterns of a smart home inhabitant. By applying suitably adapted sequential data-mining techniques to historical smart home data, consisting of an inhabitant's device interactions, we extract device usage patterns that permit us to predict each user's next action. The predicted action could then be used to send signals to the appropriate devices through the home gateway, thereby automating the home.

The unavailability of real smart home data, due to cost and privacy issues, led us to design a synthetic data generator based on discrete-event simulation, capable of generating plausible spatial-scenario-based smart home data. We used a controlled variation technique to generate similar data repeatedly so it can be used to test the data mining application. We induced temporal heterogeneity to represent time variations in day-to-day user device interactions. We also used a parameterizable Discrete Time Markov Chain (DTMC) to generate varying proportions of patterned and non-patterned smart home data. We found that our prediction system gave a useful rate of correct predictions over a wide range of tuning parameters and proportions of patterned and non-patterned data.

We strongly believe that this system will be an important component of the basic prototype platform for promoting independence to seniors and/or the physically challenged, who require assisted living to remain in their own homes.

List of Tables

5.1	Device Interaction Data for Maruthi on Monday (Week 1)	57
5.2	Device Interaction Data for Maruthi on Monday (Week 2)	58
5.3	Device Interaction Data for Maruthi on Monday (Week 3)	58
5.4	Device Interaction Data for Maruthi on Monday (Week 4)	59
5.5	Output from Synthetic Data Controller Support Bean	59
5.6	Output from Data Mining Pre Processor	60
5.7	Sequence 2 Frequent Itemsets Generated by the Feature Extractor	61
5.8	Sequence 3 Frequent Itemsets Generated the by Feature Extractor	61
5.9	Sequence 4 Frequent Itemsets Generated by the Feature Extractor	62
6.1	Predictive accuracy of the system for low support count	71
6.2	Predictive accuracy of the system for medium support count	71
6.3	Predictive accuracy of the system for high support count	72

List of Figures

1.1	Percentage of world's population over the age of 65, actual and projected [12]	3
1.2	Percentage of Canadian population over 65 years of age	4
1.3	Canadian population over 80 years of age	5
4.1	Transition Probability Matrix for Discrete Time Markov Chain	38
4.2	State diagram for Discrete Time Markov Chain	39
4.3	Architecture of Synthetic Data Generator	41
4.4	Structure of VP_DATA_LOGGER	42
4.5	Synthetic Data Generator User Interface	43
4.6	Generated Synthetic Data	44
4.7	Synthetic Data Generation Results	45
5.1	Assumed Smart Home Network	49
5.2	High Level System Architecture	50
5.3	Architecture of Intelligent Miner	55
5.4	Details of the Intelligent Miner	56
5.5	Structure of VP_MINED_FEATURE	62
5.6	Mined Features displayed from VP_MINED_FEATURE	64
5.7	Input for User Action Predictor	64
5.8	Prediction from the User Action Predictor	65

6.1 Predictive accuracy with device based mining.	72
---	----

Contents

1	Introduction	1
1.1	The Need For Home Automation	3
1.2	Scenario Illustration	4
1.3	Thesis Challenges and Goals	6
1.4	Thesis Organization	7
2	Related Work	9
2.1	Data Mining Techniques	9
2.2	Example Smart Home Projects	15
2.3	Example Smart Devices	22
2.4	Synthetic Data Generators	24
3	Problem Description and Thesis Goal	27
3.1	Problem Description	27
3.2	Thesis Goal	29

4	Synthetic Data Generator	31
4.1	Overview	31
4.2	Design Considerations	33
4.3	Design Methodology	35
4.4	Architecture and Implementation Methodology	39
4.5	Results and Discussions	43
5	Intelligent Miner and User Action Predictor	47
5.1	Overview	47
5.2	Assumed Home Network	48
5.3	The Intelligent Miner	48
5.4	Design Considerations	50
5.5	Design Methodology	52
5.6	Architecture and Implementation Methodology	54
5.7	The User Action Predictor	63
5.8	Results and Discussions	65
6	Performance Evaluation	69
6.1	Overview	69
6.2	Experimental Procedure and Results	70
6.3	Results and Discussion	73
7	Conclusions and Future Work	75
7.1	Conclusion	75
7.2	Future Work and Applications	76

Glossary of definitions

1. **Association rule** : An association rule is an expression $A \Rightarrow B$, where A and B are item sets (defined below). Given a transactional database, where each transaction T is a set of items, $A \Rightarrow B$ indicates that whenever a transaction T contains A , then it probably contains B too.
2. **Item set** : An item set is a non-empty set of items of interest from the data to be mined. An item set i is denoted by (i_1, i_2, \dots, i_n) , where i_j is an item (element).
3. **Set of large k-itemsets L_k** : (itemsets with a specific minimum level of support). Each member of this set has two fields: i) itemset and ii) corresponding support count [2].
4. **Set of candidate k-itemsets C_k** : (potentially large item sets). Each member of this set has two fields: i) item set and ii) support count [2].
5. **Support** : The support s for an association rule $A \Rightarrow B$ is the percentage of transactions in the database that contains $A \cup B$ [17]
6. **Confidence** : The confidence of an association rule $A \Rightarrow B$ is the ratio of the number of transactions that contain $A \cup B$ to the number of transactions that contain A .
7. **Sequence** : A sequence is an ordered list of itemsets. A Sequence is denoted $\prec s_1, s_2, \dots, s_n \succ$, where s_j is an item set.

8. **Subsequence** : A sequence $\prec a_1, a_2 \dots, a_n \succ$ is a subsequence of another sequence $\prec b_1, b_2 \dots, b_n \succ$ if there exists integer $i_1 < i_2 \dots, i_n$ such that $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2} \dots, a_n \subseteq b_{i_n}$.
9. **Taxonomy** : A taxonomy is a user-defined hierarchy over a collection of items.
10. **Patterned data** : Patterned data are generated according to a pre-selected/given spatial scenario.
11. **Non-patterned/noisy data** : Non-patterned/noisy data are generated using a randomly selected spatial scenario.

Chapter 1

Introduction

Smart home projects are fast emerging as an important research area. A smart home is a networked home containing various electrical and electronic devices controlled by a home gateway. The home gateway manages the devices and connects the smart home to the service providers via the Internet. There are several research projects that are being carried out on smart homes: notably Microsoft's Easy Living [49], Philips Ambient Intelligence [27], The MavHome project at the University of Texas [48], Florida University's Gator Tech Smart Home [24] and Georgia Tech's Aware-Home [6]. The vision of all smart home projects is to build an intelligent environment which will adapt to, and provide maximum comfort to, its inhabitants.

The Home Technology group in TRLabs aims to build a better smart home. Phase one of the research carried out in TRLabs had two parts: 1) develop an interoperability framework [9] which will enable communication between various devices using different standards and 2) determine the location of users within the smart home [5].

Based on the determined location information, it is possible to correlate device use with

users at various locations. These location data, combined with inferred user device interactions and time stamps collected over a period of time, will contain the user's regular device usage patterns.

We were initially interested in building a user action predictor for smart homes using data mining techniques which would form the first step to promote independence for seniors and/or the physically challenged people who require assisted living to remain in their homes. We had trouble in getting valid smart home data due to cost and privacy issues. The unavailability of data led us to think of developing a data generator for generating synthetic smart home data. There are various advantages of using synthetic data for testing and evaluating a data mining system. For example, the properties of synthetic data can be altered to represent various scenarios which are not available in authentic data. Initial study of the existing synthetic data generators led us to conclude that they are not well suited for generating realistic smart home data.

There are two major components to my thesis: 1) build a synthetic data generator, capable of generating plausible smart home data consisting of user-device interactions, and 2) build a user action predictor, which uses data mining techniques on the smart home data generated/collected to find each individual's device usage patterns. This can be used to predict a user's next device interaction.

Developing a synthetic data generator and a user action predictor for smart homes requires multi-disciplinary research. To build such a real world application, knowledge from various research domains is required, such as: data mining and database management systems, current smart home projects, pervasive and ubiquitous computing, and context aware computing.

A smart home must be context-aware to provide an unobtrusive and appealing environment filled with pervasive devices that will help it's occupants to achieve their tasks at hand easily; technology that interacts closely with its occupants to the point where such interaction become implicit [38]. To make a smart home context-aware and adaptive, it must be able to automate the devices based on the current individual(s) and context.

1.1 The Need For Home Automation

It is projected that there will be 1.2 billion people over the age of 65 worldwide in the year 2025 [13]. Nearly 70% of the world's senior population will be in developing countries and 30% in developed countries. Figure 1.1 shows the elderly demographic trends in a few developed countries and it is projected that by 2050 most of the developed and developing countries will have at least 25% of their population over the age of 65 .

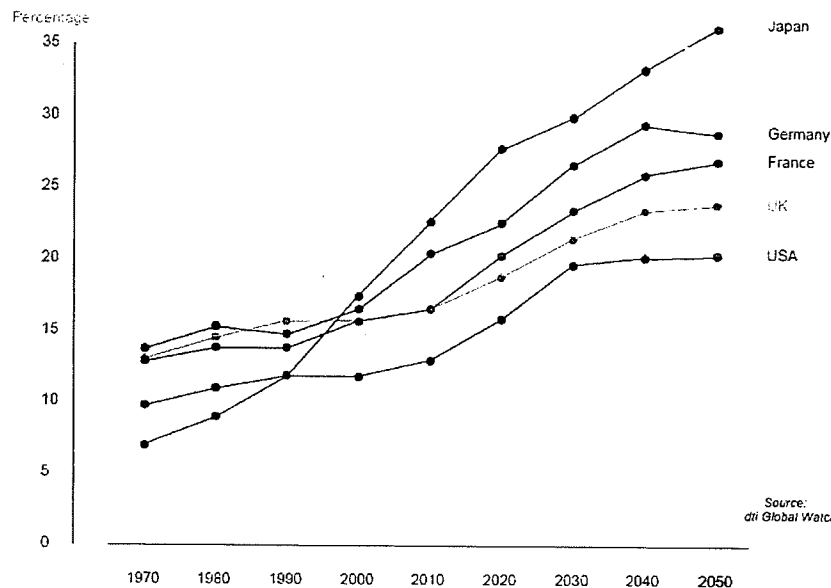
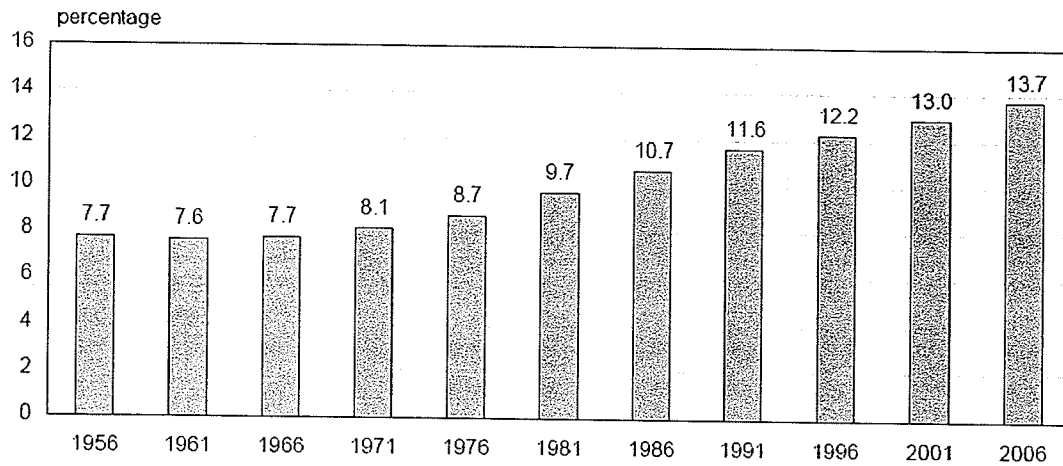


Figure 1.1: Percentage of world's population over the age of 65, actual and projected [12].

The 2006 Canadian census shows the number of Canadians, aged 65 and over increased 11.5% in the previous five years and made up a record high of 13.7% (Figure 1.2) [12] of the total Canadian population in 2006 [37].

The number of Canadians over 80 years of age has steadily increased (see Figure 1.3) [11] and reached the 1 million mark for the first time in 2006 (1.2 million) [37]. The number of centenarians in Canada went up by 22% compared with the number in 2001.

As the world's aging and physically challenged population increases, more and more people are being faced with the following questions [13]:



Sources: Statistics Canada, censuses of population, 1956 to 2006.

Figure 1.2: Percentage of Canadian population over 65 years of age

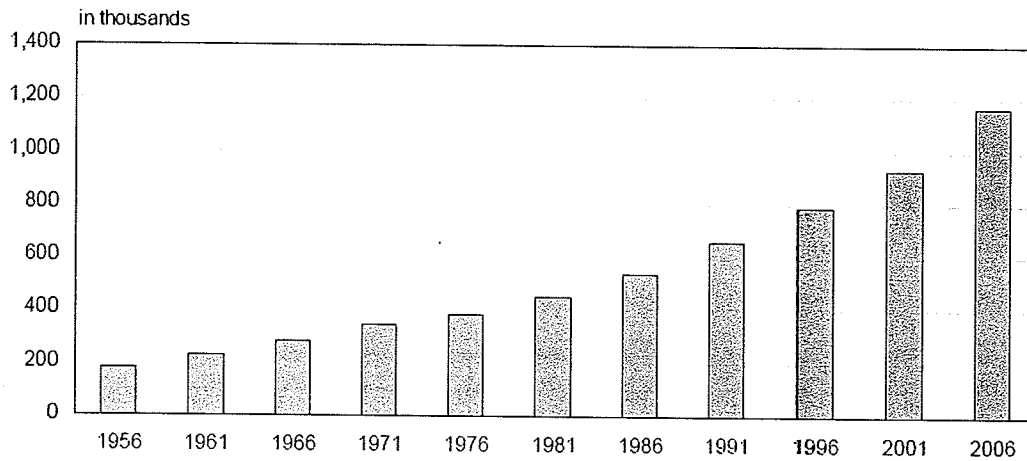
- How do you care for ailing relatives?
- How long can elderly and medically challenged individuals live alone in their own homes?

“Assistive technology” such as home automation, which can help the seniors and the physically challenged to maintain and increase their functional capabilities, is often a suitable answer to these questions.

1.2 Scenario Illustration

The need for device automation and minimization of human intervention in smart homes can be best explained with the help of scenarios [9].

It is now 6.30 AM and Maruthi is awakened by the sound of his favorite cd playing in the background. The home gateway now turns-off the air conditioner, turns-on the walkway lights



Sources: Statistics Canada, censuses of population, 1956 to 2006.

Figure 1.3: Canadian population over 80 years of age

and the coffee maker in the kitchen is turned on to make a drink. The room curtains and the blinds are opened and the bath tub is filled with warm water while the towel is warmed up in the towel heater. It is now 7.15 AM and the home gateway detects Maruthi's presence in the living room and turns on the television playing his favorite channel; at 7.30 AM Maruthi is reminded to take his daily medicines.

It is 11.00 AM and Maruthi's blood pressure and sugar levels are monitored and the readings are sent to his doctor. At 11.30 AM the doctor analyzes Maruthi's readings and finds that Maruthi's blood pressure is way above normal and hence instructs Maruthi to take blood pressure tablets. The instruction is displayed on Maruthi's TV. At 6.30 PM, the home gateway arms the interior and perimeter security system, turns on exterior and interior lights, turns-on the temperature controller, and plays Maruthi's favorite program on the television. If Maruthi is not at home, the home gateway randomizes the interior lighting (giving an appearance that the home is occupied) and records Maruthi's favorite program. It is 7.30 PM and Maruthi returns home. The home gateway detects his presence and turns-on the answering machine which plays the voice messages and turns-on the VCR to play the

recorded program.

The above scenario illustrates the advantages that the automation of devices can provide to assist smart home inhabitants.

1.3 Thesis Challenges and Goals

The main goal of this thesis is to build a useful prototype which can emulate scenarios similar, but not limited to, the one described in Section 1.2. To achieve this successfully, the following objectives, defined by Huber [25], must be met:

- optimize inhabitant productivity by improving comfort, simplifying technology use, and enhancing accessibility,
- ensure and enhance security, and minimize operational cost.

The knowledge of device interactions that occur in a smart home must be acquired, pre-processed, and applied to automate the smart devices [25]. This poses the following challenges that are addressed in my thesis:

- Capturing of device interactions
- Managing available smart home data
 - What data needs to be stored?
 - How to store all the smart home data collected?
- Extracting meaningful device usage patterns from available smart home data
 - How to extract user's device usage patterns?
 - How to decide the "interestingness" of a pattern?
 - How to best represent and store the user's inferred device usage patterns?

- Predicting the user’s next action for device automation
 - How to automate a smart device based on inferred user device usage patterns?
 - To what level should the devices be automated?

I have divided my thesis into five phases to systematically address these challenges. In phase 1 of my research, I have built a synthetic data generator for smart homes to generate device interactions within a smart home. In phase 2 of my research, I have developed a data storage module that interacts closely with the synthetic data generator to securely store the meaningful generated/collected data. In phase 3 of my research, I have developed a data miner which uses suitably adapted sequential pattern mining to extract device usage patterns on the data collected. In phase 4 of my research, I have developed a predictor, that uses the extracted device usage patterns and context to make predictions of the user’s next action. In phase 5 of my research, I have evaluated the prediction accuracy of my predictor.

1.4 Thesis Organization

The remainder of the document is organized as follows: The related work is presented in Chapter 2, and has been further divided into four major parts. In part one of the related work, I briefly explain the essential sequential data mining techniques. In part two of the related work, I describe the main smart home research initiatives. In part three of the related work, I present a short survey on some of the available smart devices. In Part four of the related work, I review existing synthetic data generators. Chapter 3 explains the challenges and the goals of my thesis. In Chapters 4, and 5, I explain the design goals and methodologies, architecture, implementation methodologies, and results of my Synthetic Data Generator and Intelligent Miner design and the User Action Predictor, respectively. Chapter 6, explains the evaluation strategy I used to assess the prototype and presents the results of our evaluation for accuracy of prediction. Finally in Chapter 7, I conclude the

thesis and specify some possible future work.

Chapter 2

Related Work

I have organized my related work into four Sections. Section 2.1 gives an overview of data mining algorithms. Section 2.2 gives an overview of the research that has been carried out in the smart home field. Section 2.3 gives an overview of some smart devices specifically designed for smart homes. Section 2.4 gives an overview of the main available synthetic data generators.

2.1 Data Mining Techniques

Data Mining can be defined as [34] “the non-trivial extraction of implicit, previously unknown, and potentially useful information from data”. Using data mining techniques, large databases can be analyzed to discover regularly occurring patterns. Data mining techniques have been applied for studying customer’s buying patterns in stores, for credit card fraud analysis, for network intrusion detection, for business process improvement, etc. However,

very little work has been carried out in applying data mining techniques to smart home data. The ability to find patterns automatically from input data can be used on smart home data consisting of location and user device interactions to find the device usage patterns of each inhabitant of a home. From the usage pattern for a user, we can predict the user's next action. Once the user's next action is predicted, it is possible to send commands to the home gateway to regulate the particular device(s) involved.

There are various data mining techniques [17] such as; association rule mining, frequent set mining, sequential pattern mining, classification, clustering and outlier detection. Of these various techniques, sequential pattern mining is of particular interest to my thesis because most device usage patterns are sequential. For example, after getting up in the morning, people might, with some degree of regularity, turn on a water heater, and then the toaster followed by turning on the coffee maker.

Understanding association rule mining will help to better understand sequential pattern mining. So I will begin by presenting a brief introduction to association rule-based mining followed by an introduction to sequential mining algorithms. The terms used in this Section are summarized in a glossary of definitions presented at the beginning of this thesis.

The Apriori Algorithm

The Apriori [2] algorithm is one of the first and most popular algorithms for association rule mining. It is based on the downward closure property or the "Apriori" property which states that any subset of a frequent itemset must also be frequent [2]. The Apriori algorithm scans the transaction database to find the frequent 1 itemset, L_K ($K = 1$) based on given values for support and confidence. The L_k ($K = 1$) set consists of two fields namely; items and their support. In stage 2 the algorithm self joins L_k ($K = 1$) to generate C_{k+1} ($K = 1$) i.e., candidate set for frequent 2 itemset. Then, the database is scanned again, to get the support count of the items in the candidate set consisting of two frequent itemsets L_k ($K = 2$). The process is repeated until all the frequent item sets in the database are generated. Drawbacks

of the Apriori algorithm include the fact that it requires multiple database scans (which are costly) and that it does not allow user focus, (i.e., the user only inputs the values for minimum support and confidence, and waits for the result). Thus, the user has no control over the mining process.

Sequential Mining Algorithms

Sequential pattern mining is the mining of frequently occurring patterns related to time or other sequences [22]. For example, consider a customer buying bread, who is likely to buy butter or jam next. This is an example of a sequential pattern. Many real-time transactions such as web access patterns or customer buying patterns are sequence and/or time related. Similarly, device usage patterns by a smart home inhabitant are also time and/or sequence related. So, it seems logical that sequential pattern mining techniques should be applicable to smart home data, to discover regularly occurring sequences that reflect inhabitant's device usage patterns. In the following subsections, I will review a few of the more common sequential mining algorithms.

The AprioriAll Algorithm

The AprioriAll algorithm was proposed for market data analysis by Srikanth and Agarwal [3]. Each transaction in the customer database consists of a customer identification number, transaction time and the item purchased. To mine the sequential data, the authors have proposed 5 phases, namely: the sort phase, Litemset phase, transformation phase, sequence phase and maximal phase.

In the sort phase, the database is sorted using the customer id as the major key and transaction time as the secondary key. This is done to convert the original transactional database to a database consisting of customer sequences [3]. In the Litemset (Large itemset) phase all the itemsets which have support greater than some minimum support are found.

The item sets are mapped to a set of continuous integers [3]. The reason attributed to this is a) comparison of any two itemsets can be done in constant time and b) it reduces the time consumed to check a sequence with that of a customer sequence. In the transformation phase each transaction is replaced with the Litemset contained in the transaction and the transactions which do not contain Litemsets are discarded. The sequence phase uses a variation on the Apriori algorithm, where frequent sequential patterns are found based on support and confidence. The maximal phase is used to find the maximal sequences among the set of large sequences. Maximal sequences are found by deleting sequences from the set of all large sequences S which are subsequences of S_k (in short, deleting sequences from $S_1 \dots S_{k-1}$ which are subsequences of S_k) [3].

The GSP (Generalized Sequential Pattern mining) Algorithm

The AprioriAll algorithm has some major drawbacks. It does not consider time constraints¹ between the transactions, taxonomies² and has a rigid definition of a transaction, i.e., all the elements of a pattern must be present in a single transaction.

In GSP [4], the user can define time constraint variables such as mingap, maxgap, sliding window³ size and is-a hierarchy. The variables mingap and maxgap defines the minimum gap and maximum time gap, in which different transactions can be considered as a related sequence of transactions respectively. The sliding window variable is defined to overcome the rigid definition of a transaction as in AprioriAll. The rest of the steps are similar to that of the AprioriAll algorithm.

¹Minimum and maximum time gaps between adjacent elements of sequential patterns.

²A user defined "is-a" hierarchy on items [4].

³Transaction time constraint, (i.e., sliding window size) determines whether a transaction contributes to the confidence of a sequence. All the transactions within the sliding window gap are considered to be a part of a single sequence.

The GSP algorithm outperforms AprioriAll in terms of efficiency because GSP generates and counts fewer candidates than AprioriAll. The effect of time constraints and the sliding window allows GSP to run significantly faster, with the maxgap parameter set to a realistic value, than without. On the other hand, specifying the sliding window increases the execution time due to the increase in overhead and increased number of sequential patterns.

The SPIRIT (Sequential Pattern mining with Regular expression constraint)

Algorithm

The Apriori based algorithms for sequential pattern mining lack user-focus (i.e., the user has no control over the mining process). Lack of user focus resulted in disproportionate computational costs and large volume of useless data [19, 20]. This serious drawback motivated Garofalakis et al. [19] to come up with a new family of sequential mining algorithms which provide user focus through the use of regular expression constraints. Compared to the Apriori based algorithms, the SPIRIT algorithm has two major architectural differences: a) weaker constraints [20] are induced by using regular expressions; and b) these weaker constraints are used in candidate generation and pruning steps. There are four variations on the SPIRIT algorithm, namely: SPIRIT (Naive), SPIRIT (Legal), SPIRIT (Valid) and SPIRIT (Regular). The various SPIRIT algorithms differ from each other based on the degree to which the regular expression constraints are enforced during the candidate set generation and pruning steps.

The regular expression constraints are advantageous because they are simple and can specify natural syntax for the sequential patterns. In a database of customer sequences, with user-specified minimum support and a user-specified regular expression constraint, the SPIRIT algorithm(s) finds the frequent sequential patterns which satisfies both the regular expression constraints and the minimum support constraint.

The use of regular expression constraints allows the user to specify the patterns of interest.

For example if the user wants to get information about hotels in New York starting from Yahoo!'s home page he can choose either of the following two paths [19]:

Travel -> Yahoo!Travel -> North America -> United States ->

New York -> New York City -> Lodging -> Hotels

or

Travel->Lodging->Yahoo!Lodging->New York->New York Cities->

New York City->Hotels and Motels

Now, consider that we are only interested in paths that begin with Travel and end in either Hotels or Hotels and motels, and at least contain one of the following; Lodging, Yahoo! Lodging, Yahoo! Travel, New York or New York City. The Garofalakis et al. [19] use regular expression constraints to extract the paths which are interesting to the user. The above problem can be easily formulated as a regular expression as follows:

Travel(Lodging|Yahoo!Lodging|Yahoo!Travel|New York|New York City)

(Hotels|Hotels and Motel),

where “|” stands for disjunction.

Using regular expression constraints and minimum support constraints results in fewer patterns which are of more interest to the user.

The SPIRIT (Valid) algorithm requires that the candidate sequences generated end in the final state of the automaton corresponding to the regular expression. The algorithm uses a stronger relaxed constraint for generating and pruning candidates (i.e., SPIRIT (Valid)

requires that every candidate sequence to be valid, meaning that the sequence path must end in the final state represented by automaton corresponding to the regular expression constraint). Also, the candidate sequences having support less than the minimum support are pruned.

The SPIRIT (Valid) algorithm is of particular interest to my thesis, and my data collection and pruning techniques will be similar to the SPIRIT (Valid) algorithm. Smart home data should consist of the device interaction data, in addition giving user locations and time stamp. My idea is to collect such user device interaction data which are of interest and valid. The details of data collection and mining are discussed in Section 5.6 of this thesis.

2.2 Example Smart Home Projects

In this Section, I will present a brief overview of various smart home projects. The smart environments developed in all the discussed research projects are only capable of automating devices based on user-set preferences and not based on user device usage patterns. "Intelligence" in these research systems are merely an application deployed [9] or preferences of individuals set in the home gateway. It is highly unlikely that a user will be willing to set all his preferences for device automation. However, none of the smart home projects except the MavHome project focuses on minimizing human involvement in operating smart home devices. The MavHome project, which also uses data mining techniques for user action prediction, is discussed in the most detail.

Microsoft EasyLiving

EasyLiving is Microsoft's prototype architecture and technologies for building intelligent environments [49]. The EasyLiving project provides a context aware environment with smart devices controlled by context brokers. The EasyLiving environment is equipped with elec-

tronic vision for user tracking and uses device independent communications and data protocols to facilitate communication between various devices. EasyLiving is Microsoft's vision for intelligent ubiquitous environments. One of the scenarios envisioned is that when a user enters a dark room, the sensors detect the user's movement and the light sensors detect that room is dark and turn on the lights. In the EasyLiving environment, if a user wants to use a device, the user has to authenticate himself/herself with the device. Once successfully authenticated, the user's details are sent to the context brokers. If the user moves within the environment, the user's movements are tracked by the sensors and reported to the context brokers. When the user tries to use any other device, the context broker re-authenticates the user with the device and thereby also allows the user to access and control other devices.

Philips Ambient Intelligence Project

The aim of Ambient Intelligence [27] project is to integrate electronic appliances into people's everyday life. The design of the system involves scenarios such as where the user is to be informed that he is getting late to the office while he is brushing his teeth [9, 27]. The Ambient intelligence project is based on user preferences but it is not completely automated. Instead, it assists the users by automating devices based on each user's preferences set.

Cisco's Internet Home

Cisco has set up an Internet home [32, 41] with state-of-the-art off-the-shelf smart devices. The residential gateway and wall-mounted control panels form the back bone of Cisco's Internet home. The residential gateway is always connected to the Internet through a high speed DSL connection and allows controlling of voice, video and data services that are to be delivered to the home. Internet home, as the name suggests can be controlled over Internet. It is possible to view live video/pictures of each room in your house and open the front entrance door when your plumber rings the door bell while your are still in the office using

your computer.

With Cisco's Internet home it is possible to control your house lighting, room temperatures, and also exterior and interior security devices over the Internet. Also the smart devices can send important messages via email to you at your office. For example, the smart refrigerator can send an email reminding that you are out of essential groceries such as milk and bread and ask for your confirmation to automatically order the groceries over the Internet.

The CUSTODIAN Project

The CUSTODIAN (Conceptualization for User involvement in Specification and Tools Offering the Delivery of system Integration Around home Networks) project [15, 18] is a smart home research initiative at Robert Gordon University. The main objective of the project is to provide a platform to design, validate and evaluate home automation systems, particularly built for people with disabilities. The CUSTODIAN project consists of visual, easy-to-use, efficient and flexible tools which can be used to simulate a smart home environment prior to their its installation.

With the CUSTODIAN project it is possible to build a smart home based on specific user needs. One of the scenarios envisioned in this project is simulating a smart home based on a "Users Needs Report" prepared by the doctor in consultation with the user who needs assistance due to disability. This simulated environment is evaluated and validated before real physical installations take place.

Gator Tech Smart Home

The Gator Tech Smart Home is a research initiative carried out at the University of Florida. One of the main goals of this research project is to assist elderly people to live independently [9, 24]. The architecture of the prototype has several layers; the physical layer, the sensor platform layer, the OSGI based middleware layer, the knowledge layer, and the context

management layer.

The physical layer is composed of various devices such as set-up boxes, TVs, heaters, lights, etc., and their inter-connections. The sensor layer consists of sensors capable of communicating between various devices and with the OSGi [7] based middleware. The knowledge layer determines the available services and for this it uses a reasoning engine [9, 24]. The context manager activates certain services based on events. For example, reminding people to take medicines once they have had their dinner [9].

The University of Texas MavHome Project

The MavHome project aims to build a smart home that is intelligent and versatile [50]. The goal of the project is to: maximize comfort and productivity of its inhabitants and minimize operation cost [50]. Research has been carried out in the MavHome project to investigate the use of data mining algorithms for user action prediction. They have come up with 4 different algorithms for user action prediction, they are:

1. SHIP : Smart Home Inhabitant Prediction.
2. TMM : Task based Markov Model.
3. ED : Episode Discovery.
4. Predict : Meta-predictor.

SHIP

The history of inhabitant actions is maintained as a sequence of actions. The SHIP algorithm then tries to match the recent sequence [16] of actions with historical data. The user's actions are constantly mapped to the history. A match queue is maintained that gives near linear

runtime⁴ performance. The SHIP algorithm has two stages of operation: a) updating the match queue, b) evaluation of the matches, and predicting the next user action.

In the first stage, there are two operations. Whenever the system detects a new action the history is updated. First, the length of the longest sequence at time t $l_t(s, a)$ that ends with action a in state s [16] is matched with the historical sequence immediately prior to time t . The authors also define a parameter frequency measure $f(s, a)$ that represent the number of times the action a has been taken from the current state. In the second stage, based on the values of match length and frequency from stage 1, the matches in the queue are evaluated.

The algorithm is based on sequence prediction, i.e., it can only predict the next event in the user's device usage sequence and not the time at which it will occur. This is a limitation of the SHIP algorithm and this assumption will not hold true in a real life scenario. In a real life scenario devices must be automated based upon both the time of device usage and user device interaction patterns to achieve good accuracy.

The designers of the SHIP algorithm claim that SHIP yields a predictive accuracy of 94.4% on synthetic data and 53.4% on real data [16]. But, these results only show that the SHIP algorithm can predict the user's next action based on the user's historical device interaction sequence but not the time at which the action needs to be performed. For this reason, the real world applicability of SHIP seems to be limited.

TMM - Task-based Markov Model

The main reason for using a Markov model, is the need to learn the pattern of user actions without storing large amounts of historical data. In this approach, the a user action is represented as a string such as;

A:10/25/2005 8:15:32 PM Kitchen Light D1 on.

This string can be represented as a single state model. The same operations with minimal

⁴A match queue consists of user's common device usage sequences.

time difference can be merged into a single state in the model. Whenever an action is recorded, it is first checked whether it matches any of the states in the previous models. If it does, within an acceptable time gap, then it is merged with the existing state. If there is no match, then a new state is created in the model. Then, based on the Markov model, it is possible to predict the user's next action. By matching the current action of the user with one of the previously built models, the next action can be predicted as the outgoing transition from that state having the greatest probability.

The authors claim that if the actions that comprise a task can be identified, then it is possible to identify the current task and generate more accurate transition probabilities for the corresponding task [48]. To achieve this, the authors have partitioned the action sequence into smaller sub-sequences that are part of the same task. The partitioning results in a set of groups which can be clustered into similar tasks.

The Episode Discovery (ED) Algorithm

The Episode Discovery algorithm is based on the sequential mining algorithm developed by Srikant and Agarwal [4]. The ED algorithm identifies significant episodes within an inhabitant event history. A significant episode can be viewed as a related set of device events that may be ordered, partially ordered or unordered [23]. Examples of episodes are:

heat on (daily)⁵, sprinkler on (weekly)⁶, etc.

The data is mined by partitioning the input sequence into transaction-like collections of events by sliding a window over the event history. ED then evaluates the potential sequences using the Minimum Description length (MDL) technique. The MDL is a technique where the description length of the database is minimized by replacing each instance of the pattern with a pointer to the pattern [23]. Using the MDL, the ED algorithm will predict the user's next action.

⁵Heater is turned on daily.

⁶Sprinkler is turned on weekly.

Predict

The Predict (meta-predictor) algorithm takes input from other algorithms (SHIP, ED, TMM etc.), and uses a backpropagation neural network to learn a confidence value for each of the above mentioned prediction algorithms. The confidence value is calculated based on historical data gathered and the accuracy of each of the algorithms on this data. Meta features like number of inhabitants in the home, the training time, number of devices in the environment etc., are also considered for calculating the confidence value of each of the prediction algorithms. Final predictions are based on a voting scheme dependent upon weighted votes from each individual prediction algorithm.

In summary, the research carried out in the MavHome project focuses on device automation in smart homes using various techniques such as: task based Markov models, association rule mining and sequential pattern mining.

Most prediction algorithms used in MavHome have on an average 70% – 90% predictive accuracy on synthetic data, and have 20% – 60% on real life data.

One of the main concerns of these prediction algorithms is the data on which these algorithms were tested. We don't know the quality of the synthetic data generated to test these prediction algorithms. Also, the data used for each of the prediction algorithms seems to be different, raising concern that the data generated might be biased in favor of the particular algorithm.

The MavHome prediction algorithms show good accuracy for synthetic data but their accuracy of prediction on real smart home data is marginal. This clearly shows that there is a wide discrepancy between the synthetic data generated and the real smart home data. Also, the MavHome's synthetic data generator has not been well explained (to best of my knowledge). So, I assume that the synthetic data generator used in MavHome for testing these prediction algorithms is of low quality.

2.3 Example Smart Devices

In this Section, I present an overview of a few of the available smart devices, which can be easily integrated in a smart environment. Smart devices can be either controlled manually or by the home gateway.

Gate Reminder

The Gate Reminder [31] is a smart system located near the front door of a smart home. It reminds users what they have to know and take before leaving the home. For example wallet, cell phone, Identification card, grocery list and list of appointments. This system is particularly beneficial for people suffering from memory related problems; however it is also quite useful for senior people as well as the general population.

DigiFlower

DigiFlower [46] is a location-based smart system which is capable of tracking the family members in a smart home. The DigiFlower consists of digital flowers each of which represent a family member. It can detect a family member returning home and the corresponding flower of that family member blossoms in the DigiFlower frame. The authors envision that this smart device will provide emotional satisfaction to the family members and will be a new interface for communication among the users.

Smart Bed

Smart Bed [46] system is aimed to wake you up in a pleasant ambiance based on your preference settings. It can be configured to play your preferred music, produce the smell of

coffee and so on from about thirty minutes before the actual alarm time.

Smart Pillows

Smart Pillows [46, 30] are designed to personalize your bedtime. They can play your favorite music and read bed time stories or even books. Once you start sleeping the smart pillow can automatically check the quality of your sleep and will gradually decrease the volume and turning off eventually. Smart Pillows can check your body temperature, blood pressure level, pulse and respiration at regular intervals and, if there is a need, it will automatically report to your hospital through the Internet.

Smart Refrigerators

Smart Refrigerators [30, 10] are equipped with inventory application. These smart refrigerators are capable of keeping track of the individual food items based on their RFID tag. It can track the expiry dates of the food items and also the quantities of food items. If it detects an expired food item it will notify the user. It can even prepare shopping lists [30] for the food items to buy once they are consumed. Once this shopping list is prepared, it will notify you and wait for your confirmation for placing the order in a grocery store of your choice or the one that is nearest to your home.

The various smart devices discussed in this Section can seamlessly integrate and publish their services to the home gateway. The home gateway can start controlling these devices based on the registered services. With the availability of smart devices the complexity of integrating, configuring, and automating devices is reduced.

2.4 Synthetic Data Generators

Synthetic data generators are used by many researchers and application developers to test the performance of their prototypes on synthetic data representing real life scenarios. Most of the data mining algorithms are tested using synthetic data. Using synthetic data has five advantages: 1) It gives flexibility on the type of data created e.g XML, text, etc. 2) Often, the real data is not available or is very hard to get due to privacy issues and/or cost factors. 3) The amount of data generated can be controlled. 4) Various scenarios, for which real life data is not available, can be emulated. 5) Synthetic data generated can provide necessary traction for newer data mining techniques and approaches that otherwise might never be thought off [28].

In this section, I summarize some of the research that has been done in synthetic data generators. There are various commercial data generators available such as GSApps's GS Data generator [21], SQL Edit's DTM Data Generator [51], and CoSort's Rowgen [14]. All these data generators have a limitation as identified by Jeske et al. [28]. These systems generate data by simply assuming the data to be independent and none of them consider, or are aware of, the often complex relationships between the various data attributes.

SDG - A System for Synthetic Data Generation

SDG [8] is a system that generates text data for different problems having similar structure, solution type, and level of difficulty. This is done to assist students synthesize problems on their own. The system can generate synthetic data of linear structure. The system requires the user to provide a template called the SDtemplate written in SDscript based on which the synthetic data will be generated. This SDtemplate is interpreted by the SDInterpreter to generate synthetic data called synthetic data objects. The SDInterpreter generates different text data (SDObjects) randomly. The SDInterpreter reads the SDtemplate and generates new texts or strings, or discards existing lines and strings.

SDG generates data based on templates and not based on events. It is important to note

that all the device interactions within a smart home are discrete events, so the architecture of SDG will not support generation of smart home data. Further, it is not capable of repeatedly generating similar data so, SDG cannot be used to test a data mining application. Also SDG is template based, so it does not have the intelligence to generate complex smart home data.

IDSG - IDAS⁷ Data and Scenario Generator.

Jeske et al. [29] describe the IDSG architecture. The authors cite research that shows realistic or accurate data generation is a hard problem. Lin et al. [35], consider various implicit and explicit rules governing the generation of synthetic data. To incorporate these rules, they categorize the rules into 3 types. Type 1 are the independent rules; an example for this is the rule specifying number of hours in a day. Type 2 are the Intra-record (horizontal) rules. The horizontal rules can relate one or more, or all, attributes of a record together [35]. An example of a horizontal rule is generation of an individual's income based on his/her education, age, profession, and gender. Type 3 are the Inter-record (vertical) rules which define the relationships among the various records; an example is the total number of records of individuals living a town.

The IDSG uses semantic graphs for representing relationships among data attributes and to define data generating rules. IDSG uses statistical and rules-based algorithms to generate data. The IDSG was primarily built to generate data and scenarios for personal information and credit card transactions.

IDSG generates data based on rules and not based on events so, architecturally IDSG cannot generate smart home data. Also, IDSG uses rules and not hierarchies to generate data. To generate plausible smart home data it is desirable to use hierarchies (explained in Section 5.2). Further, the authors have not validated their system, so the quality of the data is unknown.

⁷Information Discovery and Analysis Systems.

Other Synthetic Data Generators

Abounnaga et al. [1] have developed a synthetic data generator capable of generating complex-structured XML data. The system is capable of allowing high level of control over the characteristics of the generated data. The generator is flexible. It takes several input parameters that control the generated data. The authors claim that all the input parameters have simple and intuitive meanings [1]. It is important to note that this system generates data which can be used to test the performance of XML database management systems and not data mining algorithms, hence it cannot be used.

The IBM XML generator [26] is similar to work done by Abounnaga et al. [1]. Lundin et al. [36] developed a synthetic fraud data generation methodology. The unique feature of this methodology is that it takes authentic data as the base to generate synthetic data. The authors identify the important characteristics of authentic fraud data and generate synthetic fraud data based on these observed characteristics.

None of these data generators can generate plausible smart home data mainly due to their underlying architecture and the way they define a transaction. We need to have a synthetic data generator for generating smart home data consisting of user device interactions considering the fact that the device usage data within a smart home is not purely random.

Chapter 3

Problem Description and Thesis Goal

3.1 Problem Description

Smart homes will be a reality in the near future. Most smart home research efforts are aimed at building a smart environment which adapts to its users. The need for this is supported by Nielsen [42] who states that most usability research focuses on human-centered perception of devices, and is re-enforced by Norman Donald [43] who said “I’m a technology enthusiast annoyed by the unnecessary complexity of today’s products. My goal is to humanize technology, to make it disappear from sight, replaced by a human centered, activity-based family of information appliances, that are easy to learn, easy to use, powerful and enjoyable”. An important task in making an environment smart and human centric is to be able to make predictions of what its users might want to do next. By predicting the user’s next device interaction, devices can be automated, thus adapting to the user’s needs and simplifying device use.

A smart home consists of various appliances with varying computational capabilities and functionalities. These devices are usually operated manually and their computational capabilities are not utilized fully. To provide maximum comfort to the inhabitants and to exploit these smart devices, I propose to support the automation of these smart home appliances based on the inhabitant's device usage patterns. The device usage activities of the smart home inhabitants such as, turning on the air-conditioner, television, coffee maker or heater, can be collected/generated over a period of time. The data thus collected/generated, contains extensive knowledge of the smart home inhabitant's device usage patterns. Data mining techniques can be used to extract the commonly reccuring device usage patterns and to use them to automate the various smart home devices.

The device usage pattern of a smart home inhabitant varies based on the device capabilities. Smart home appliances can be categorized into a) low functionality devices such as: lights, coffee maker; b) medium functionality devices such as: air-conditioners, heaters, etc; c) high functionality devices such as: televisions, VCRs, and DVD players. Also the data from various smart home devices vary based on their computational capabilities. For example, the data from a low functionality device such as a light will consist of a time stamp, a user id, and the device on-time and off-time; whereas the data from a high functionality device, such as a television, will consist of user id, time stamp, on time, off time, channels viewed, and duration of view for each of the channels watched. I propose that combining user-based mining and device-based mining (based on device computational capabilities) will improve the accuracy of the prediction made.

My research focuses on solving three problems:

- 1) Building a synthetic data generator based on discrete event simulation, which is capable of producing plausible smart home data;
- 2) Building an intelligent data miner that can appropriately apply adapted sequential data mining techniques to the smart home data to extract each inhabitant's device usage pattern.
- 3) Designing scenario based experiments to evaluate the capability of the miner to predict

inhabitant's next device interaction based on extracted device usage patterns by altering the amount of patterned and non-patterned data (noise) generated synthetically.

3.2 Thesis Goal

My goal is to build a prototype simulation of a smart environment, which is user centric, adapts to its inhabitants and yet is simple to use. This system will form the basic prototype platform for promoting independence to seniors and/or the physically challenged, who require assisted living to remain in their own homes. I strongly believe that my research will further the smart home technologies and will be of potential social value.

This page is intentionally left blank.

Chapter 4

Synthetic Data Generator

4.1 Overview

We were initially interested in building an action predictor using data mining techniques which would promote independence for seniors and/or the physically challenged who require assisted living to remain in their homes. We had trouble in getting valid smart home data due to cost and privacy issues. The unavailability of data led us to think of developing a synthetic data generator for smart home data. There are various advantages of using synthetic data for testing and evaluation of a data mining system. The properties of synthetic data can be altered to represent various scenarios which are not available in authentic data. Initial study of the existing synthetic data generators led us to conclude that they are not well suited for generating accurate synthetic smart home data.

Generating plausible smart home data is inherently complex due to the various options available to generate them. Smart home data generated may be time/device based, inhab-

itant based and/or spatial scenario based. Time-based data generation means that data is generated based on the time of the device interactions. Device-based data generation means that the data is generated based on the characteristics of devices. Inhabitant-based data generation means that the data is generated based on individual's device usage sequences. Spatial-scenario-based data generation means that the data is generated using the spatial location of individuals in the home.

Each of these methods individually is not capable of producing realistic and plausible smart home data, further adding to the complexity. Device/time-based data generation can produce illogical data. For example, device-based data generation can produce illogical data such as; *a user in a washroom using the water heater at 7.00 AM. and simultaneously turning on the garage light.* Similarly, time-based data generation can also result in illogical data being generated such as *a user operating a coffeemaker in the kitchen at 7.00 AM and then operating the same coffeemaker again at 7.15 AM .*

To overcome these inconsistencies, we have developed a hierarchical synthetic data generator combining spatial scenarios at level one and device interactions at level two. The spatial scenario forms the basis for synthetic data generation i.e. our data generation will be based on spatial location of the user and the available devices in that particular location. By using this hierarchical scheme it was possible to overcome the limitations of both device-and/or time-based data generation. We induced time heterogeneity to represent time variations in day-to-day user/device interactions. Our data generator is capable of generating device interaction data for a particular inhabitant and not for multiple inhabitants in a single run. However the data generation simulator can be run multiple times for generating distinct data for each of the multiple inhabitants. Our data generator is capable of generating varying proportions of patterned (consistent data) and non-patterned data (noise), a useful feature to test the accuracy and efficiency of our data miner.

While developing our synthetic data generator we primarily had seven goals to be met:

- To develop a stochastic data generator which will be capable of producing plausible smart home data,

- Capable of producing similar data repeatedly,
- Capable of producing varying amounts of patterned and non-patterned data so as to test for false positive and false negative results of our action predictor,
- Capable of producing time-heterogeneous day-to-day device interactions,
- Capable of producing synthetic data based on either device capabilities and/or real smart home data (when/if such data becomes available),
- Easily configurable and fast enough to produce the larger amounts of data required to test a data mining algorithm, and
- Interoperable with various commercially available database management systems.

The restrictions, assumptions and limitations of our data generator are discussed in Section 4.5.

4.2 Design Considerations

The user's device interactions within a smart home can be viewed as events. For example turning on a light is an event; turning on a Television is an event. So, our first design consideration was to generate data based on events; for this we use an event-based simulator (SSJ) to produce synthetic data. To produce logical data, we wanted the generated data to be based on user's location and devices within that particular location (e.g., a room). The other design consideration we had was to generate different user-device interaction data for weekdays and weekends; this is done to represent a more realistic situation. While this heterogeneity has been characterized as weekend and weekdays, the system is not limited to this; for example, Mondays and Fridays can also have different routines. The underlying assumption is that there must be day-based recurring patterns. This is, of course, commonly reflected in real world situations.

Further, to add flexibility, the data generator must be capable of determining the pre-determined spatial scenarios randomly or accepting them as an input. Within each of the spatial locations, the data generator must be capable of selecting the device interactions randomly. To test a data mining application, it is required to generate similar data repeatedly with varying proportions of noisy data. This formed the basis of the second and the third design consideration; the data generator must be capable of generating varying proportions of patterned and non-patterned data repeatedly.

It is logical to incorporate time variability in the user-device interactions because it is not possible for a user to operate any particular device at exactly the same time everyday. The fourth design consideration was to incorporate time variability in the user-device interactions and this representation of variability must be made consistent with real data once made available. It was decided to use multiple device interaction files having the same device interactions but the time of these device interactions were varied a little (typically ± 10 minutes).

Smart home appliances based on their computational capabilities can be categorized into: a) low functionality devices such as, lights, coffee maker, etc. b) medium functionality devices such as air-conditioners, heaters, etc. and c) high functionality devices such as televisions, VCRs, DVD players, etc. The data generated varies based on the device capabilities. Any data generated must contain the following information: the user who used it, the location of the user, and location of the device. Apart from this information, the data also consists of device specific data; for example, a low functionality device such as coffeemaker, lights etc., will have only two states of operation: when the device was turned on, the duration of operation. The data from a medium functionality device such as air conditioner will have times of usage, duration of usage, and such device-specific data as temperature and humidity level maintained. Data from a high functionality device such as a TV will have the time of usage, duration, and potentially much more device-specific data such as the channel watched, volume level, brightness level, contrast level and color level, etc.

The other design consideration was to make the data generator capable of generating

synthetic data based on real data if it was made available. This was the fifth design consideration.

To build a useful and efficient synthetic data generator, it is also important that the data generator be easily configurable with available commercial database management systems such as Oracle [45], Microsoft SQL [39] server, and also open source database systems such as MySQL [40], etc. The synthetic data generator must be capable of producing huge volumes of data quickly and efficiently.

4.3 Design Methodology

Our data generator is a discrete event simulator, generating data based on the spatial location of the user and the devices available within that location. The spatial location can either be given by the user or selected randomly from the spatial scenario text file. Seeds of the random number generator can be changed to produce different streams of random numbers and, in turn, different sets of spatial scenarios can be selected using these random numbers. An example spatial scenario string will have the following format,

```
MB*~LR*~KI*~MBB*~LR
```

The above string represents the movement of a user within the home, the user is in the master bedroom then he goes to the living room followed by the kitchen then the master bedroom bathroom and then back to the living room. Within each of these locations, the user's device interactions are generated differently (randomly) or similarly (systematically) by selecting the devices within that particular location from the device interaction file. The number of locations within a single spatial scenario string is defined as its length. In our experiments we limited the length of a spatial scenario string to 5. Once the real smart home data is made available the spatial scenarios can be extracted based on any one or both of the following two ways 1) based on time : for example, morning 7-00 AM to 9-00 AM. 2)

based on a condition: for example, from the time the user enters the home to the time he leaves the home.

The random numbers generated for selecting the device interactions can be varied by changing the seeds used for their generation. This will lead to generating different device interaction data in that particular spatial location for various replications. To generate similar (systematic) data over various replications, the same seeds are used in generation of random numbers, which are used to select the device interactions within the given spatial location. An example entry in the device interaction file will be of the form,

```
LF%L1,07-15,5*~MF%AC,07-16,120,TEMP=20
```

The above string indicates that a low functionality device (light 1) was turned on at 7-15 AM and the duration in on-state was 5 minutes and then the AC was turned on from 7-16 AM to 9-16 AM and the temperature was maintained at 20 C.

The time variation was achieved by manually keying in the event times and storing them in the different device interaction files. From these sets of different device interaction files, the system randomly chooses one of the device interaction files, which have similar device interactions but the events are time varied.

Apart from the time variation there are two other sources of variability in the device interactions sequences that are induced manually;

1) Skipped device interactions: For example, the events chosen from a device interaction file for Monday on week 1 might have the following events

```
LF%L1,07-15,5*~MF%AC,07-16,120,TEMP=20
```

and the events chosen from another device interaction file for Monday on week 2 might have the following event only

```
LF%L1,07-15,5
```


2) Reordered device interactions within a spatial location: For example, the events chosen from a device interaction file for Monday on week 1 might have the following event order

LF%L1,07-15,5*~MF%AC,07-16,120,TEMP=20

and the events chosen from another device interaction file for Monday on week 2 might have the following event order

MF%AC,07-14,120,TEMP=20*~LF%L1,07-15,5

This process of manually changing the a) start and/or end times of device interactions b) skipped device interactions and c) reordered device interactions, was chosen to correspond closely to the recording of real device interactions. It was thought that this was a more realistic and flexible mechanism than using some scheme of random device interaction generation.

To test a data mining application, we needed to produce similar (patterned) data repeatedly with varying proportions of noisy (non-patterned) data. To generate patterned smart home data, we used a controlled variation technique where the same seeds are used for generating random numbers that choose a specific spatial scenario over multiple replications. A Discrete Time Markov Chain was incorporated, to alternate between patterned and non-patterned data producing states. The Discrete Time Markov Chain uses a 2×2 probability transition matrix to switch between two states 1) patterned data generating state and 2) non-patterned data generating state.

In the patterned data generating state, the same, spatial scenario from Spatial Scenario text file and device interactions within each spatial location are chosen to generate data. The repeating device interactions were chosen from various device interaction files, where the start times and end times of these device interactions were varied. There are three sources of variability in the patterned data generation 1) The start and end times of device interactions 2) Reordered device interactions within a spatial location and 3) Skipped device interactions

within a spatial location. All the three sources of variability were injected manually by using multiple device interaction files for each day of the week as described earlier.

In the non-patterned data generating state, the spatial scenarios are selected randomly and mostly different from the spatial scenario used in the generation of patterned data. The amount of patterned and non-patterned data produced can be configured by changing the transition probabilities. We used DTMC because it provides additional source of variability in a reasonably realistic way.

	Patterned Data	Non-patterned Data
Patterned data	P_x	$1-P_x$
Non-patterned Data	$1-P_y$	P_y

Figure 4.1: Transition Probability Matrix for Discrete Time Markov Chain

A general 2X2 probability transition matrix is shown in Figure 4.1 and the corresponding Discrete Time Markov Chain is shown in Figure 4.2. The DTMC in Figure 4.2 can be understood as follows; when the DTMC is in State 1 (patterned data generating state) the probability that it will stay in the same state is given by P_x and the probability of a change in state to state 2 (non-patterned generating state) is given by $1 - P_x$. Similarly, P_y is the probability it will continue to stay in state 2 once in state 2 and $1 - P_y$ is the probability representing the change of state from state 2 to state 1. By controlling the values of P_x and P_y it was possible to produce varying proportions of patterned and non-patterned data. The transition takes place after the completion of the data generating process for the previously selected spatial scenario.

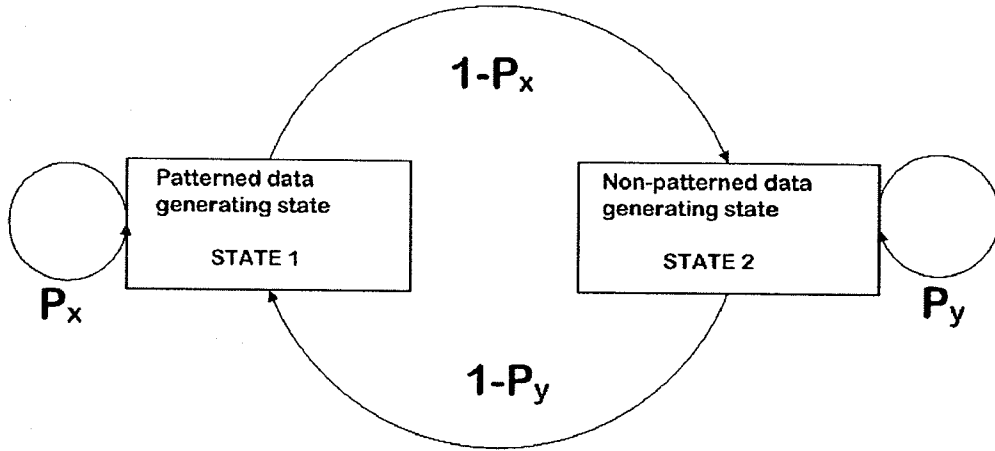


Figure 4.2: State diagram for Discrete Time Markov Chain

Different text files were used to generate data on weekdays and weekends. This unique feature is not seen in any of the existing data generators. By using various text files for different days of the week, the data generator is made flexible and realistic.

Our synthetic data generator is easily configurable as it uses text files. There are various advantages of using text files. Real smart home data can be stored in these text files and based on these real life data, synthetic data can also be produced. Finally, text files are easy to manage, inspect and transfer.

With minimal configuration, the synthetic data generator is also capable of generating data and storing the data in various commercially available databases such as Oracle [45], Microsoft SQL [39] and MySQL [40].

4.4 Architecture and Implementation Methodology

The architecture of our synthetic data generator is shown in Figure 4.3. The Synthetic Data Controller collects data from the user through the Text User Interface. The user keys in

information, such as home gateway id, user id, start date, end date, and optional spatial scenario sequences. Based on these inputs the Synthetic Data Controller will either generate data based on the user given spatial scenario or it generates a random key and using this random key it selects a spatial scenario from the spatial scenario text file. Provisions were made to change seeds for generating different random keys. This provides a way to perform independent replications for generating smart home data.

Once the spatial scenario has been selected by the Synthetic Data Controller, the control of the system is transferred to the Synthetic Data Generator. The Synthetic Data Generator, based on each spatial location extracted for the spatial scenario will call the Random Key Generator and Retriever. The Random Key Generator and Retriever, generates a random key and selects the particular device interaction sequence based on the spatial location of the user and returns this data back to the Synthetic Data Generator. The Synthetic Data Generator will pass the data generated to the Database Controller, which sets up the necessary communication between our system and a third party database management systems based on the DB Configuration file. Once the communication channel is set up, the generated data is stored in the appropriate Database system. The Date Converter Utility plays an important role in converting various date formats to date with time stamp value and converts this value to various different formats used by the different third party database systems.

The Synthetic Data Generator was implemented using the Oracle 10g [45] database management system and with the SSJ [33] simulation system. The structure of the primary database table VP_DATA_LOGGER, used to store our generated data is shown in Figure 4.4.

In the database, The GATEWAY_ID field stores the home gateway number, the USER_ID field stores the user identification number, the USER_LOCATION field stores the user's current location, the DEVICE_CAP field stores the device capability information (such as low, medium and high functionality devices), the DEVICE_ID field stores the id of the particular device being used, and the START_DATE_TIME and END_DATE_TIME store the start date, start time, end date and end time of a particular device interaction. The

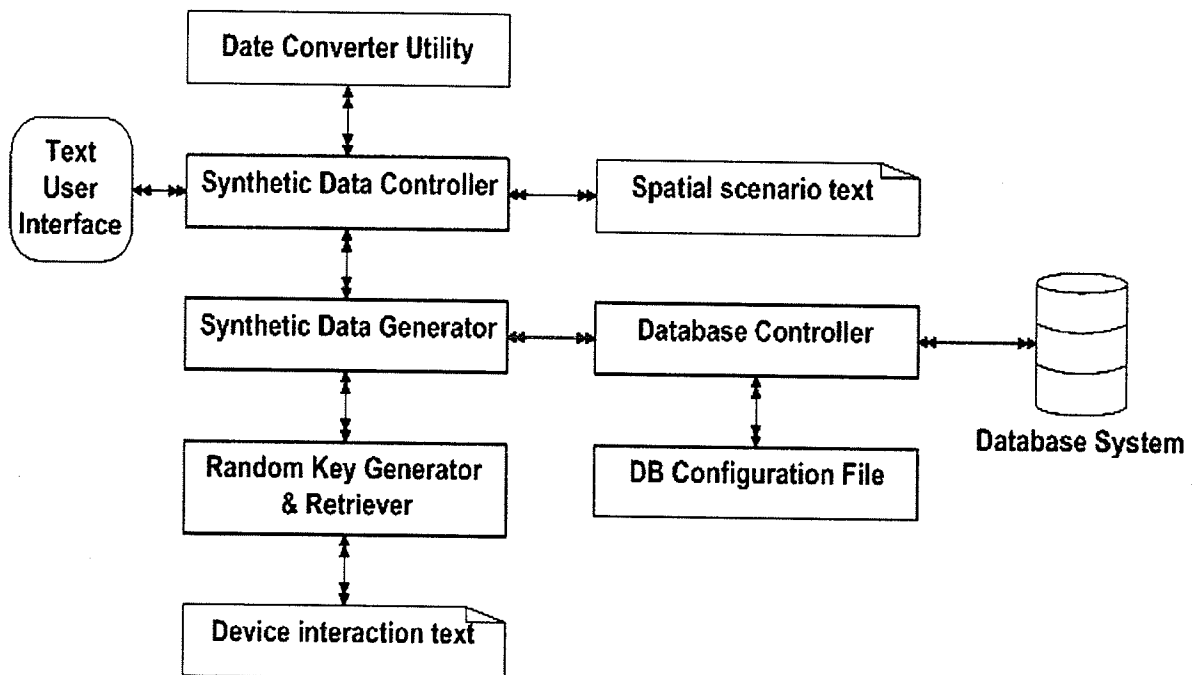
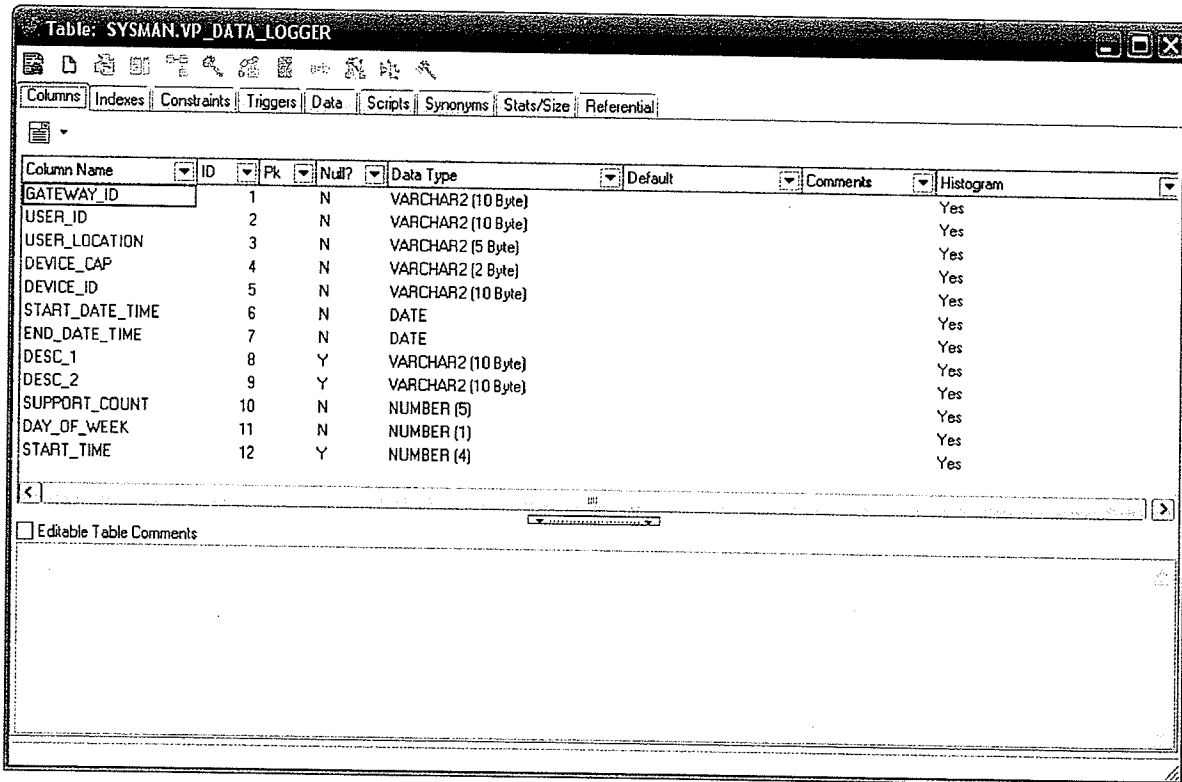


Figure 4.3: Architecture of Synthetic Data Generator

DESC.1 and DESC.2 fields are used to store additional information for high and medium functionality devices. For example generated data for a TV interaction might consist of additional information such as channel watched, volume, brightness, color and contrast level maintained, which will be stored in the DESC_1/2 fields.

The SUPPORT_COUNT, DAY_OF_WEEK and START_TIME are specialized fields used for data mining as described in Chapter 5. For a new device interaction the SUPPORT_COUNT starts at 1 and for each identical device interaction, triggered within a predefined time-gap, the SUPPORT_COUNT is increased by one. The DAY_OF_WEEK field stores the day of the week, in which the device interaction has occurred. This is done to uniquely identify the day-to-day activities of the smart home inhabitants (more details in chapter 5).



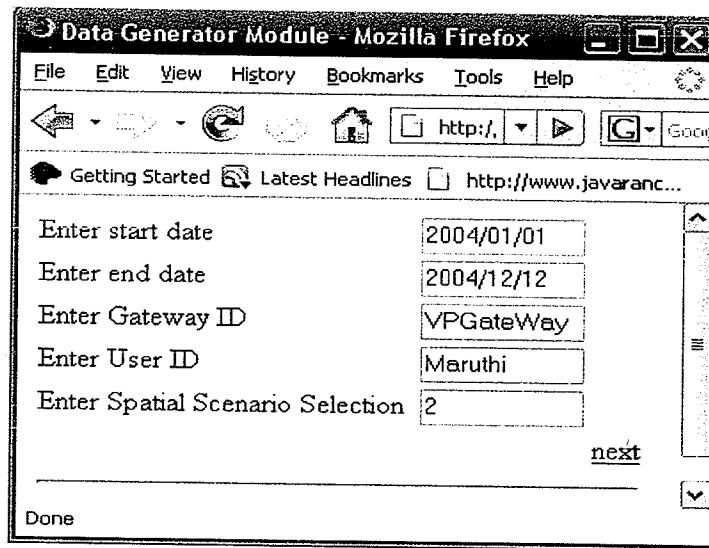
Column Name	ID	Pk	Null?	Data Type	Default	Comments	Histogram
GATEWAY_ID	1		N	VARCHAR2 (10 Byte)			Yes
USER_ID	2		N	VARCHAR2 (10 Byte)			Yes
USER_LOCATION	3		N	VARCHAR2 (5 Byte)			Yes
DEVICE_CAP	4		N	VARCHAR2 (2 Byte)			Yes
DEVICE_ID	5		N	VARCHAR2 (10 Byte)			Yes
START_DATE_TIME	6		N	DATE			Yes
END_DATE_TIME	7		N	DATE			Yes
DESC_1	8		Y	VARCHAR2 (10 Byte)			Yes
DESC_2	9		Y	VARCHAR2 (10 Byte)			Yes
SUPPORT_COUNT	10		N	NUMBER (5)			Yes
DAY_OF_WEEK	11		N	NUMBER (1)			Yes
START_TIME	12		Y	NUMBER (4)			Yes

☐ Editable Table Comments

Figure 4.4: Structure of VP_DATA_LOGGER

4.5 Results and Discussions

The Synthetic Data Generator met the design goals and an efficient implementation led to generate and store 773 synthetic data records/sec on average. Our synthetic data generator is capable of producing approx 2.2 million records per hour on a dual core AMD Athlon 64 bit system with one gigabyte of RAM. This rate of production will allow us to generate a large number of sample interactions having various characteristics quickly so they can be used as input for our data mining experiments. Figure 4.5 illustrates the data generator's user interface. Figure 4.6 and Figure 4.7 illustrate the sample data generated along with statistics on the proportion of valid (patterned) and invalid (non-patterned) data produced, respectively.



The screenshot shows a web browser window titled "Data Generator Module - Mozilla Firefox". The address bar shows "http://www.javaranc...". The page content includes a form with the following fields and values:

Field	Value
Enter start date	2004/01/01
Enter end date	2004/12/12
Enter Gateway ID	VPGateWay
Enter User ID	Maruthi
Enter Spatial Scenario Selection	2

Below the form is a "next" button. The status bar at the bottom shows "Done".

Figure 4.5: Synthetic Data Generator User Interface

We have developed a synthetic data generator for smart homes meeting the design considerations identified in this chapter. The Synthetic Data Generator is quick, capable of producing plausible smart home data and can be easily configured with commercially available database management systems. The data generator is only capable of producing data

Table: SYSMAN.VP_DATA_LOGGER

Columns Indexes Constraints Triggers Data Scripts Synonyms Stats/Size Referential

Sort by Primary Key

GATEWAY_ID	USER_ID	USER_LOCATION	DEVICE_CAP	DEVICE_ID	START_DATE_TIME	END_DATE_TIME	DESC_1	DESC_2	SUPPORT_COUNT	DAY_OF_WEEK	START_TIME
VPGateway	Maruthi	BS	MF	AC	01/01/2004 8:19:00 AM	01/01/2004 9:16:00 AM	TEMP=22		50	2	819
VPGateway	Maruthi	KI	LF	CM	01/01/2004 6:15:00 AM	01/01/2004 6:21:00 AM			50	2	615
VPGateway	Maruthi	BD2B	MF	HT	01/01/2004 6:05:00 AM	01/01/2004 6:20:00 AM	TEMP=50		50	2	605
VPGateway	Maruthi	BD2	MF	AC	02/01/2004 8:18:00 AM	02/01/2004 9:16:00 AM	TEMP=22		50	3	818
VPGateway	Maruthi	LR	MF	AC	02/01/2004 8:20:00 AM	02/01/2004 9:16:00 AM	TEMP=22		100	3	820
VPGateway	Maruthi	BS	MF	AC	02/01/2004 8:19:00 AM	02/01/2004 9:16:00 AM	TEMP=22		50	3	819
VPGateway	Maruthi	KI	LF	CM	02/01/2004 6:15:00 AM	02/01/2004 6:21:00 AM			50	3	615
VPGateway	Maruthi	BD2B	MF	HT	02/01/2004 6:05:00 AM	02/01/2004 6:20:00 AM	TEMP=50		50	3	605
VPGateway	Maruthi	BD2	MF	AC	03/01/2004 8:18:00 AM	03/01/2004 9:16:00 AM	TEMP=22		50	4	818
VPGateway	Maruthi	LR	MF	AC	03/01/2004 8:20:00 AM	03/01/2004 9:16:00 AM	TEMP=22		100	4	820
VPGateway	Maruthi	BS	MF	AC	03/01/2004 8:19:00 AM	03/01/2004 9:16:00 AM	TEMP=22		50	4	819
VPGateway	Maruthi	KI	LF	CM	03/01/2004 6:15:00 AM	03/01/2004 6:21:00 AM			50	4	615
VPGateway	Maruthi	BD2B	MF	HT	03/01/2004 6:05:00 AM	03/01/2004 6:20:00 AM	TEMP=50		50	4	605
VPGateway	Maruthi	BD2	MF	AC	01/01/2004 8:18:00 AM	01/01/2004 9:16:00 AM	TEMP=22		50	2	818
VPGateway	Maruthi	LR	MF	AC	01/01/2004 8:20:00 AM	01/01/2004 9:16:00 AM	TEMP=22		100	2	820
VPGateway	Maruthi	BD2	MF	AC	04/01/2004 8:18:00 AM	04/01/2004 9:16:00 AM	TEMP=22		49	5	818
VPGateway	Maruthi	LR	MF	AC	04/01/2004 8:20:00 AM	04/01/2004 9:16:00 AM	TEMP=22		98	5	820
VPGateway	Maruthi	BS	MF	AC	04/01/2004 8:19:00 AM	04/01/2004 9:16:00 AM	TEMP=22		49	5	819
VPGateway	Maruthi	KI	LF	CM	04/01/2004 6:15:00 AM	04/01/2004 6:21:00 AM			49	5	615

Row 1 of 35 total rows

Figure 4.6: Generated Synthetic Data

based on individual users at a particular time; it does not generate data for multiple users at the same time; however interactions for multiple users can be created by combining the results of multiple runs. In future, it will be possible to extend the base data generator to support multiple user data. To generate multiple user data it is important to consider independence and non-independence between device interactions of different users. We also plan to make this data generator client - server based, where any external system can send requests to our data generator.

```

Start Server - C:\bealuser_projects\domains\WPMINER1_2new\startWebLogic.cmd
Start date and time 2004/01/01
Days Gap 346.0
The user id Maruthi
The First Level Random Number Generated 0.8640137675020061
The spatial scenario selected BD2*~LR*~BS*~K1*~LR*~BD2B
Connection Successful
The 10 th number is < This is to done to prove that controlled variation is being achieved for data type selection > :0.23630425314216053
Synthetic Data Generation Results
*****
REPORT on Tally stat. collector ==> Propotion of generated USHD< Valid Smart Home Data >
=====
min      max      average  st. dev'n  Base (num obs)
=====
0.00     1.00     0.89     0.32      2076

95.000000 percent confidence interval for mean: ( 0.87 , 0.90)
REPORT on Tally stat. collector ==> Propotion of generated ISHD< Invalid Smart Home Data >
=====
min      max      average  st. dev'n  Base (num obs)
=====
0.00     1.00     0.11     0.32      2076

95.000000 percent confidence interval for mean: ( 0.10 , 0.13)
The Synthetic Data SyntheticDataGenerator Object com.vpminer.sdg.SyntheticDataControllerBean@22775d

```

Figure 4.7: Synthetic Data Generation Results

This chapter marks the end of my first and second phases of research, where I have addressed the following two of the four challenges originally identified in the Section 1.3,

- Generating of device interactions, and

- Managing available smart home data.

Chapter 5

Intelligent Miner & Action Predictor

5.1 Overview

In this chapter I present the assumed smart home environment, followed by detailed description of the Intelligent Miner¹ and the User Action Predictor². At the end of this chapter, I present the identified limitations of the Intelligent Miner and the User Action Predictor.

¹The Intelligent miner applies data mining techniques on available smart home data to extract device usage patterns of an particular inhabitant.

²The User Action Predictor is responsible for predicting a inhabitant's next action based on extracted device usage patterns.

5.2 Assumed Home Network

Figure 5.1 depicts the assumed smart home network, consisting of many devices such as televisions, fridge, lights, air-conditioner, heaters, coffee-maker and smart sofa connected to a home gateway possibly using various protocols and wired/wireless technologies. In our prototype home network we have the home gateway³ connected to the smart home service provider's servers through high speed Internet connection. This is similar to the Home Area Network [44] considered by Pourreza [47]. Most of the smart devices (except low computational devices) available in the market, have the capability to register their services with the home gateway. The home gateway can use these registered services to control various devices. We consider that the configuration of a typical home gateway might range from 64 to 128 Mb of memory with a 400 - 600 MHz on-board processor. Most of the research [49, 48, 9] carried out in smart home projects fail to acknowledge this fact. They use powerful desktop PC's as home gateways. It is unlikely, due to cost and reliability constraints in the real world, that a powerful desktop computer will be used as the home gateway. Further, I assume that the home gateway has the necessary logic to record and to map the start and the end times of particular device interactions and send them to the remote server located at the service provider's end. The mining of the raw smart home data takes place in these remote servers.

5.3 The Intelligent Miner

The Intelligent Miner forms the core of our system. The raw smart home device interaction data are preprocessed and mined by the Intelligent Miner, which finds and extracts the regular device usage patterns of an individual. Figure 5.2 shows the high level architecture

³The home gateway device, apart from controlling and connecting various devices within the home, connects the home network to the service provider through high speed Internet connections.

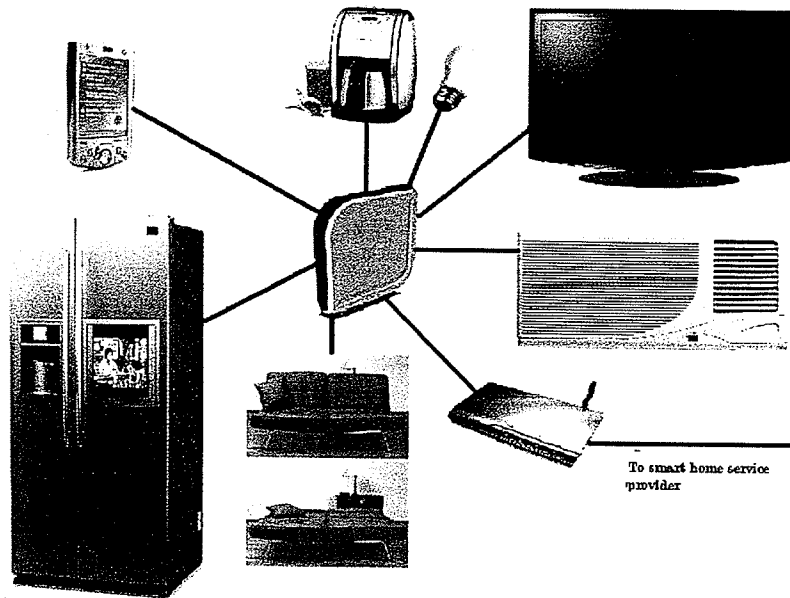


Figure 5.1: Assumed Smart Home Network

of the mining system with many home gateways connected to the data storing and data mining servers at the service providers' end.

While developing our Intelligent Miner we set the following design goals and conditions:

- Mining process to be done by powerful servers located at the service providers end.
- Use appropriate data collection and pruning techniques on the collected smart home data.
- Use an appropriate data mining technique, adapted to suit a smart home environment.
- Find device usage patterns based on a particular individual and the day of the week.
- Find device usage patterns based on a particular device.
- Collect and store data from the home gateway either in plain text format or XML format.

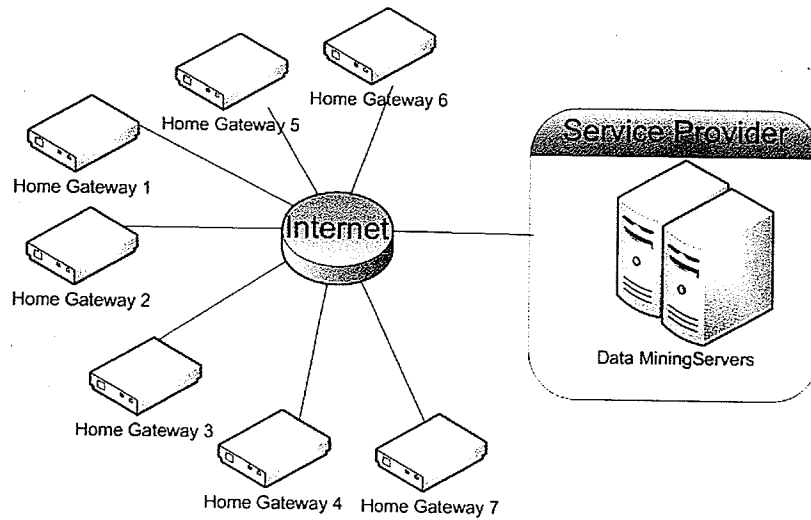


Figure 5.2: High Level System Architecture

- Send the predictions made back to the home gateway either as plain text or in XML format.

5.4 Design Considerations

We assume that the configuration of a typical home gateway might range from 64 to 128 Mb of memory with a 400 – 600 MHz on-board processor. With this configuration, the home gateway alone cannot store all the device interactions and mine the collected data. To overcome this challenge, we decided to have a server powerful enough to store the huge amounts of smart home data and mine it. This is reflected in the first design consideration.

I wanted to collect only the device interaction data that are valid, i.e., events which are significant and noticeable (described in Section 5.5). Collecting all the smart home device interaction increases the load on the database servers and decreases the mining algorithm's efficiency because of the huge amount of useless data that needs to be processed. This was

the second design consideration.

Many real-time transactions such as, web access patterns or customer buying patterns are sequence and/or time related. Similarly, device usage patterns by a smart home inhabitant are also time and/or sequence related. For example, after getting up in the morning, people might, with some degree of regularity, turn on the water heater and then the toaster followed by turning on the coffee maker, etc. To find sequential time-related patterns of events, it was decided to use altered sequential mining techniques. This was the third design consideration.

The aim is to find the individual device usage patterns rather than those of finding common device usage patterns of a population. In general sequential pattern mining, the sequential patterns are found based on population/global data and not based on a particular individual. General sequential pattern mining is not of great use in a smart home scenario, because we are more interested to know about the preferences of an individual and his/her device interaction patterns rather than that of a population. It is only based on the individual's device usage patterns that it is possible to make predictions of what the individual is going to do next. In order to find sequential patterns of a particular individual, it was required to alter the existing sequential pattern mining algorithm to mine data based on the individual and the day of the week. This was the fourth design consideration.

In a few situations it is important to understand the overall device usage, rather than individual usage. For example, in a house occupied by 2 people, both of whom work, the husband might turn on the coffeemaker at 7.00 am for a few days and on some other days the wife might turn on the coffeemaker at 7.00 am. In such cases, if the data are mined based on users alone, we will not be able to effectively predict overall device usage. However, considering the combined usage of these devices by various inhabitants within a time period might lead to an interesting pattern. Similarly, it will be possible to find the most popularly viewed TV program by the family. Supplementing user-based mining with device-based mining was the fifth design consideration.

It was also desired that the Intelligent miner to be flexible enough to handle different data input types and capable of giving output data in various formats particularly in plain text

and XML format, because most home gateways either send and/or receive and/or handle plain text or XML data. These were the sixth and seventh design considerations respectively. There are various advantages of using XML data such as, standardized data formats, easy to parse, and easy to transfer data between various systems.

5.5 Design Methodology

Most of the commercially available gateways have enough computational power to run simple applications such as a client web browser, etc. A simple web browser application is used in the system emulation to communicate with the remote server. The web browser client can send and/or receive data to/from the server.

The generated/received data between the server, and the home gateway will consist of user device interaction data. The data will be a string such as:

```
GAT_ID=100*~USER_ID=G10001*~ST_TIME=15.00.00*~APP_ID=TV*~CHAN=273*~
```

```
END_TIME=15.30.00*~ACTION_FLAG=T
```

The above string contains the following information: the request is from a home gateway with device id 100, the user id is G10001, the device with which he interacts is TV, the channel he watches is 273, and he has watched the program from 3.00 PM to 3.30 PM.

Our pruning technique is based on the ACTION_FLAG; the data will be stored only if the ACTION_FLAG is true. The criteria for the ACTION_FLAG is determined in the following way: Is it an event, if so is the event significant?

An event is an interaction by the inhabitant with a device. Even if an event occurs, it is important to determine whether it is a significant event or not. A significant event is one where the user device interaction is there for a minimum period of time. We have made this minimum time gap a parameterizable value. For, example turning on the television and

browsing a few channels does not form a significant event. However watching a particular channel for more than 15 minutes forms a significant event. Only if the data is significant will it be recorded. This pruning technique, based on significant events, is similar to regular expression constraints used in SPIRIT algorithm [19] and this reduced the amount of data we collected, stored, and mined. The time duration for considering a device interaction as significant is a parameterizable value. This value can be replaced with acceptable range of values when the real data is made available.

The normal sequential-pattern mining algorithms only consider the time between events and not the duration for which a particular event occurs. This was the first design alteration we had to do to the existing sequential-pattern mining algorithm to suit a smart home scenario.

Device usage patterns by a smart home inhabitant are time and/or sequence related. General sequential mining algorithms take into account the time-gap between events. Based on this time-gap, the algorithm determines whether a particular event is in a sequence or not. But in the smart homes scenario, we cannot have a definitive time-gap between the events. For example, the time between device interactions might vary from two minutes to more than three hours. So we had to change the existing sequential mining algorithm, so as to not take into account the time gap between device interactions. This was the second design alteration we did in the existing sequential-pattern mining algorithm to suit the smart home scenario.

To predict what the user's next action is, it is necessary to mine data based on individual users. Existing sequential mining algorithms are not designed to mine based on individuals; rather, they find out common patterns among a population. This was the third design alteration we did to the existing sequential mining algorithm to perform pattern mining based on individual users and the day of the week, which suits to the smart home environment.

Also to improve the accuracy of the predictions made, we used device based mining as a fall back mechanism. The predictor makes predictions based on mined features and if there are no predications available for the particular input data, the predictor queries the database

for a prediction based on overall device usage i.e., it tries to find whether there is a device which has been used commonly at that time by many different users and if found, the device is automated (as explained Section 5.4).

5.6 Architecture and Implementation Methodology

The Home Gateways are assumed to be connected to the server through the Server Gateway. The server stores the inhabitant's preference and device usage data. The data collected is sent to the Data Parser. The Data Parser takes an input XML string and, based on the function code, routes the extracted data in `<name, value>` pairs to the Data Logger. The Preference Manager⁴, will store the user preference to a data base. The Preference Manager will also allow editing of user preference.

The Data Logger logs only valid device interaction data to the database⁵. The Intelligent Miner will mine the data collected, using our altered sequential pattern mining algorithm described later in this Section. Finally the second XML parser receives the data containing device usage patterns for each of the inhabitants, converts this into an XML or Text string and sends it back to the Home Gateway through the Server Gateway.

The Intelligent Miner

The Intelligent Miner is responsible for mining the data collected for each user in order to find his/her device usage patterns. In our system, the Intelligent miner is responsible for:

- finding the support count of each inhabitant's valid device interaction sorted by user id, day of the week and the device interaction start and end times,

⁴A provision for taking into account the possible use of user preference in the future.

⁵A valid device interaction has the ACTION_FLAG as true.

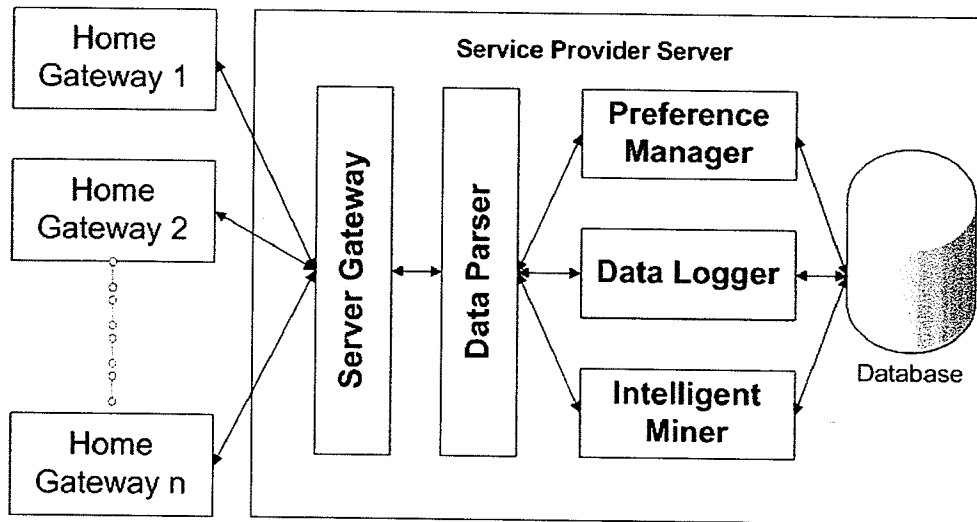


Figure 5.3: Architecture of Intelligent Miner

- finding frequent one itemset consisting of device interactions which have the required/predefined support count,
- finding, iteratively, the candidate set for frequent-2 itemsets by self-joining frequent-1 itemsets and passing over the database to find the support count of the generated candidate-2 itemsets and find the frequent-2 itemsets. The miner repeats the joining process till it finds all the sequences.

The details of the algorithms are described later in this Section. Since we only want individual device usage patterns based on the days of week, the data will be mined based on user id, day of the week and the device interaction start and end-times. This method has the following advantages:

- The number of candidate sets generated will be fewer than what is generated by the original algorithm; this reduces the computational load on the mining algorithm and on the database servers.
- The data is mined based on user id and day of the week. So, the mining process is

independent from each of the other users and different days of the week. Because of this it will be possible in the future to parallelize the mining process based on two characteristics; one based on user id and the other based on the day of the week for each user.

- Mining data, associated with an individual and day of the week, will give accurate device usage patterns for that individual on specific days of the week and hence enables improved accuracy of the predicting system.

Figure 5.4 shows the internal structure of the Intelligent Miner. The Intelligent Miner depends on the Synthetic Data Controller Support Bean, the Data Mining Pre Processor, the Data Mining Feature Extractor, and the Extracted Feature Displayer to perform data mining and display the extracted/mined feature.

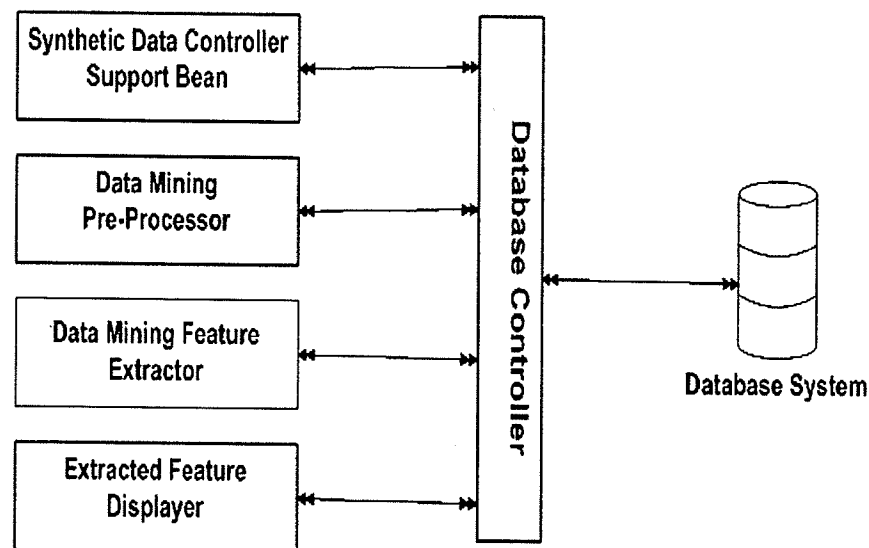


Figure 5.4: Details of the Intelligent Miner

The following scenario will be used to illustrate how the Data Mining Feature Extractor interacts with the Synthetic Data Controller Support Bean, and the Data Mining Pre Processor to find an individual's sequential device-usage pattern.

Table 5.1: Device Interaction Data for Maruthi on Monday (Week 1)

Device ID	Start Time	End Time	Duration	Device Capability	Mapped to
Coffee Maker	7:16 am	7:30 am	14 minutes	LF	1
Washroom Light	7:18 am	7:28 am	14 minutes	LF	2
Television	7:30 am	7:45 am	15 minutes	HF	3
Water Heater	7:35 am	7:46 am	11 minutes	MF	4
Basement Washroom Light	7:50 am	8:00 am	10 minutes	LF	5

It is now Monday morning 7:15 am and Maruthi wakes up. He then turns on the coffee maker at 7:16 am. He turns on the washroom lights at 7:18 am. He leaves the washroom and turns off the washroom light at 7:28 am. It is now 7:30 am and Maruthi turns on the television and watches channel 41 where CNN is aired. At 7:30 am he turns off the coffee maker. At 7:35 am he turns on the water heater, he then turns off the television at 7:45 am and he also turns off the water heater at 7:46 am. He then turns on the lights in the basement washroom at 7:50 am and takes a bath till 8:00 am, and when he has finished taking his bath, he turns off the light and leaves.

We vary this sequence of events for a four week period, and run through the mining algorithm. Table 5.1, Table 5.2, Table 5.3, and Table 5.4 represent Maruthi's device interactions on each Monday over a four week period (Note the variations in the start times and the end times of the device interactions in Tables 5.1, 5.2, 5.3 and 5.4 and also note the reduced device interactions in Table 5.3).

The Synthetic Data Controller Support Bean is responsible for updating the support count of each repeating device interaction. It works on the VP_DATA_LOGGER table shown in Figure 4.4. Each device interaction generated/received is recorded in the VP_DATA_LOGGER table. The Synthetic Data Controller Support Bean is responsible for two operations:

- 1) When a device interaction is generated/received, the Synthetic Data Controller Support Bean first checks the database for a similar interaction based on the user, on the given

Table 5.2: Device Interaction Data for Maruthi on Monday (Week 2)

Device ID	Start Time	End Time	Duration	Device Capability	Mapped to
Coffee Maker	7:18 am	7:33 am	15 minutes	LF	1
Washroom Light	7:22 am	7:30 am	8 minutes	LF	2
Television	7:32 am	7:50 am	18 minutes	HF	3
Water Heater	7:35 am	7:46 am	11 minutes	MF	4
Basement Washroom Light	7:52 am	8:00 am	8 minutes	LF	5

Table 5.3: Device Interaction Data for Maruthi on Monday (Week 3)

Device ID	Start Time	End Time	Duration	Device Capability	Mapped to
Coffee Maker	7:20 am	7:33 am	13 minutes	LF	1
Washroom Light	7:24 am	7:32 am	8 minutes	LF	2
Television	7:35 am	7:50 am	15 minutes	HF	3
Water Heater	7:36 am	7:47 am	11 minutes	MF	4

Table 5.4: Device Interaction Data for Maruthi on Monday (Week 4)

Device ID	Start Time	End Time	Duration	Device Capability	Mapped to
Coffee Maker	7:19 am	7:34 am	15 minutes	LF	1
Washroom Light	7:23 am	7:31 am	8 minutes	LF	2
Television	7:33 am	7:51 am	18 minutes	HF	3
Water Heater	7:36 am	7:47 am	11 minutes	MF	4
Basement Washroom Light	7:53 am	8:01 am	8 minutes	LF	5

Table 5.5: Output from Synthetic Data Controller Support Bean

Mapped Device Interaction 1-Sequences	Support Count
1	4
2	4
3	4
4	4
5	3

day of the week, on the predefined acceptable variation in the start times of the device interactions, on the user's location, and on the device.

2) If there are no matches, it logs the interaction as a new event and with a support count of 1. If there is an exact match or a match with acceptable variability in the start time of the device interaction, the Synthetic Data Controller Support Bean updates the respective support_count field in the VP_DATA_LOGGER to support_count+1.

When the Synthetic data controller is supplied with data from the Tables 5.1 to 5.4, it performs the two operations to process the input data. The result of this operation is shown in Table 5.5.

Table 5.6: Output from Data Mining Pre Processor

Mapped Device Interaction 1-Sequences	Support Count
1	4
2	4
3	4
4	4

The Data Mining Pre Processor is responsible for extracting all the frequent-1 itemsets from the VP_DATA_LOGGER table based on predetermined support count given by the user. The Data Mining Pre-processor identifies the device interactions which have the required support count and records them in the VP_DATA_PROCESSOR table. The structure of VP_DATA_PROCESSOR table is identical to the structure of the VP_DATA_LOGGER shown in Figure 4.4, the only difference being that VP_DATA_PROCESSOR has only the valid frequent-1 itemset of device interactions.

For example, let us take the support count as 4, then the Data Mining Pre Processor works on the data in Table 5.5 and the resulting data is shown in Table 5.6.

The Data Mining Feature Extractor is responsible for executing the altered sequential mining process. The Data Mining Pre Processor works on the VP_DATA_PROCESSOR to extract regularly occurring sequential device interactions and stores the extracted features in the VP_MINED_FEATURE table. The Data Mining Feature Extractor finds the candidate-2 itemsets by self joining the frequent-1 itemsets. It then has a read-only pass over the database to find the actual support count of the sequences in the generated candidate-2 itemset. The actual frequent-2 itemsets are then generated from the candidate-2 itemset based on a predefined support count. The result of this process is shown in Table 5.7. The algorithm repeats this process to find frequent-3 itemsets (Table 5.8), frequent-4 itemsets, etc; till there are no more candidate itemsets to be generated. Table 5.8 shows the final and the longest mined frequent-4 itemset.

Table 5.7: Sequence 2 Frequent Itemsets Generated by the Feature Extractor

Mapped Device Interaction 2-Sequences	Support Count
<1,2>	4
<1,3>	4
<1,4>	4
<2,3>	4
<2,4>	4
<3,4>	4

Table 5.8: Sequence 3 Frequent Itemsets Generated the by Feature Extractor

Mapped Device Interaction 2-Sequences	Support Count
<1,2,3>	4
<1,2,4>	4
<1,3,4>	4
<2,3,4>	4

The extracted device interaction sequences are stored in the VP_MINED_FEATURE table. The structure of VP_MINED_FEATURE is shown in Figure 5.5.

The extracted Feature displayer is a simple utility program which displays the mined/extracted features in the browser application as shown in Figure 5.5.

Table 5.9: Sequence 4 Frequent Itemsets Generated by the Feature Extractor

Mapped Device Interaction 2-Sequences	Support Count
<1,2,3,4>	4

Table: SYSMAN.VP_MINED_FEATURES

Column Name	ID	Pk	Null?	Data Type	Default	Comments	Histogram
GATEWAY_ID	1	N	N	VARCHAR2 (10 Byte)			Yes
USER_ID	2	N	N	VARCHAR2 (10 Byte)			Yes
DAY_OF_WEEK	3	N	N	NUMBER (1)			Yes
MINED_FEATURE	4	N	N	VARCHAR2 (100 Byte)			Yes

☐ Editable Table Comments

Figure 5.5: Structure of VP_MINED_FEATURE

5.7 The User Action Predictor

The User Action Predictor is responsible for making predictions of what the user is going to do next, based on a user's present location, time, and day of the week.

The system can give any one of the following three types of prediction:

- 1) Correct prediction 2) Wrong prediction 3) No prediction.

We find the spatial scenario and the device interaction sequences for which the patterned data was generated. The input to the User Action Predictor is generated by the Synthetic Data Generator, the output from the Predictor is compared to the patterned spatial scenario and the device interaction sequences found, if there is a match it is taken as a correct prediction or else it is taken as a wrong prediction. A generated device interaction is said to match a previously mined device interaction if the start and the end times of these two device interactions are within an acceptable time-gap and if all the other ancillary features are exactly similar. For example, the ancillary features to TV might have the channel watched, the volume and contrast level maintained etc.

If there are two possible predictions with exactly the same support count, then the User Action Predictor returns a no prediction as the result. For, evaluation purposes both the no prediction and the wrong prediction are considered to be wrong predictions.

The data input screen for the User Action Predictor is shown in Figure 5.7. As we can see from Figure 5.7, the request for prediction is from gateway having an id GVP100 for user VP and the time is 6:10 am on the 1st day of the week and VP's present location is master bedroom and VP's latest significant device interaction was with light 1 in the master bedroom.

Based on this input data and recent device interactions sent by the home gateway, the User Action Predictor queries and parses the VP_MINED_FEATURES table. It tries to match the input either exactly or partially to the mined sequence of events in the VP_MINED_FEATURES table. If a match is found, the User Action Predictor then finds the next device interaction in the sequence and sends it back to the home gateway, which in turn automates the working of the device.

Gateway ID	User ID	Day of the week	Mined Feature
GVP100	VP	1	BD2-MF-AC-TEMP=22--818~*
GVP100	VP	1	BD2B-MF-HT-TEMP=50--605~*
GVP100	VP	1	KI-LF-CM---615~*
GVP100	VP	1	BS-MF-AC-TEMP=22--819~*
GVP100	VP	1	LR-MF-AC-TEMP=22--820~*
GVP100	VP	2	KI-LF-CM---615~*
GVP100	VP	2	BD2B-MF-HT-TEMP=50--605~*
GVP100	VP	2	BD2-MF-AC-TEMP=22--818~*
GVP100	VP	2	LR-MF-AC-TEMP=22--820~*
GVP100	VP	2	BS-MF-AC-TEMP=22--819~*
GVP100	VP	3	BD2B-MF-HT-TEMP=50--605~*
GVP100	VP	3	KI-LF-CM---615~*
GVP100	VP	3	BS-MF-AC-TEMP=22--819~*
GVP100	VP	3	BD2-MF-AC-TEMP=22--818~*
GVP100	VP	3	LR-MF-AC-TEMP=22--820~*

Figure 5.6: Mined Features displayed from VP_MINED_FEATURE

Enter Gateway ID

Enter User ID

Enter Day Of Week

Enter Current Time

Enter User Location

Enter Previous Action

[next](#)

Figure 5.7: Input for User Action Predictor

The prediction made by the User Action Predictor is shown in Figure 5.8. The string from the User Action Predictor informs the home gateway that the user might turn on the coffee maker in the kitchen at 6:11 am. The home gateway would then use this information to automate the respective device.

5.8 Results and Discussions

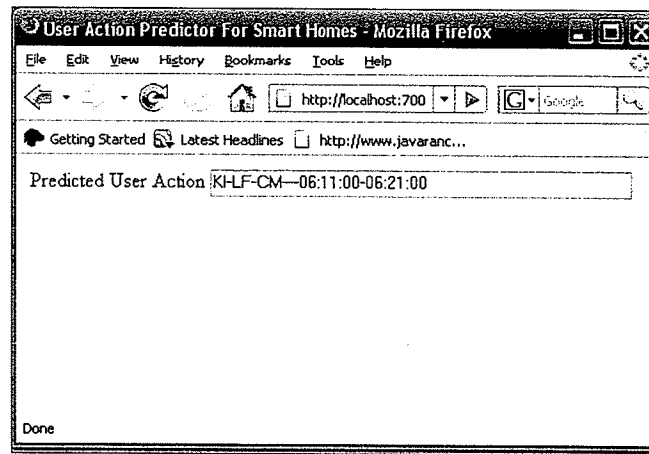


Figure 5.8: Prediction from the User Action Predictor

The Intelligent Miner and the User Action Predictor met our design goals. I have developed an Intelligent Miner for smart homes meeting the required design considerations and the required alterations to the existing sequential mining algorithm so as to make them applicable for use on smart home data, as described earlier in this chapter.

The accuracy of the User Action Predictor is dependent on the mining algorithm. In my evaluation of the Mining algorithm and the User Action Predictor, I have provided the Intelligent Miner with varying proportions of patterned and non-patterned data generated by the Synthetic Data Generator. I then checked the capability of the Intelligent Miner to

extract features from these input data sets and assist the User Action Predictor based on the number of correct predictions made. The details of the evaluation and the results are presented in Chapter 6 (Evaluation) of this thesis.

The prototype system is greatly dependent on the server. This results in centralized control which is a disadvantage. However, considering the home gateway's limited computational capabilities, we substantiate the need for using server-based services.

Also, since the home gateway needs to be always connected to the Internet in order to communicate with the server, a failure in the Internet connection can lead to stalling of the system.

Because of personal data being stored, there are privacy issues that needs to be addressed. These issues are beyond the scope of this thesis. It would be a valuable future work to address these identified external issues.

The Intelligent Miner generates many frequent itemsets, but due to the improved data pruning and collection technique based on the ACTION_FLAG, the number of candidate itemset generated is reduced, thus reducing the mined frequent itemsets compared to traditional sequential data mining algorithms. Also the generated frequent items are useful to do prediction based on partial matching.

The User Action Predictor heavily depends on the server and has to communicate to the server many times. However, there will be no major bandwidth issues because of the extremely small amount of data sent back and forth to and from the server. Also, at present the user action predictor does not calculate the confidence level for the predictions made. However, it would be possible to find the confidence level for each of the predictions made in the future by using association rules (see glossary Section). The home gateway can then determine weather to automate devices based on the particular prediction and the corresponding confidence level.

This chapter marks the end of my third and fourth phases of research, where I have addressed the following two of the four challenges identified in the Section 1.3,

- Extracting meaningful device usage patterns from available smart home data, and

- Predicting the user's next action for device automation.

This page is intentionally left blank.

Chapter 6

Performance Evaluation

6.1 Overview

In this chapter, the evaluation plan used to assess the prototype system is described in detail. The main aim is to evaluate the User Action Predictor's accuracy, which forms the performance metric, for varying proportions of patterned and non-patterned smart home data. If the predictive accuracy of the system is more than 80%, it will be assumed that the system's performance is good ; and if the predictive accuracy is more than 60% and less than 80%, it will be assumed that the system's performance is moderate but needs fine tuning. If the predictive accuracy is less than 60%, it will be assumed that the system's performance is poor.

The percentage predictive accuracy is calculated as follows:

$$\frac{\text{Number_of_correct_predictions}}{\text{Number_of_total_predictions}} * 100\%$$

The following three control parameters were used for in the study:

- 1) The proportion of generated patterned and non-patterned smart home data.
- 2) The minimum support-count of the data-mining algorithm.
- 3) The spatial scenario.

The range of control parameters were arbitrarily chosen. The proportion of patterned and non-patterned data was varied from 50% to 90% in steps of 10%. For each particular proportion of patterned and non-patterned data, three different support count levels namely low, medium and high were used:

- 1) Low : calculated as 60% of maximum possible support count.
- 2) Medium : calculated as 75% of maximum possible support count.
- 3) High : calculated as 90% of maximum possible support count.

The maximum possible support count depends on the number of weeks for which the data are generated. For example, if the synthetic data is generated for a 4-month period, since the miner works based on days of the week, the maximum support count for a device interaction will be 16 (total of 16 weeks, so a device interaction for a particular day of the week can occur up to a maximum of 16 times). Performance evaluation was assessed across replications using different spatial scenarios as described in Section 6.2.

6.2 Experimental Procedure and Results

The following two sets of experiments were conducted,

- 1) Calculation of predictive accuracy of the system when assisted with device-based mining.
- 2) Calculation of predictive accuracy of the system when not assisted with device-based mining.

Each of the two mentioned experiments were replicated ten times. Each replication was done with different spatial scenarios. For each of the spatial scenarios (replications), the proportions of patterned and non-patterned device interaction data generated was varied.

Table 6.1: Predictive accuracy of the system for low support count

Percentage of Patterned data	Predictive accuracy with device assisted mining	Predictive accuracy without device assisted mining
90%	100.00%	100.00%
80%	98.33%	98.33%
70%	86.67%	83.33%
60%	61.67%	58.33%
50%	45.00%	41.67%
Average	78.33%	76.33%

Table 6.2: Predictive accuracy of the system for medium support count

Percentage of Patterned data	Predictive accuracy with device assisted mining	Predictive accuracy without device assisted mining
90%	96.67%	96.67%
80%	95.00%	95.00%
70%	83.33%	80.00%
60%	68.33%	65.00%
50%	55.00%	50.00%
Average	79.67%	77.33%

For each of the proportions of patterned and non-patterned data, three different support counts namely high, medium, and low were used to mine the data. For each of these three different support counts six predictions were made. The predictive accuracy was calculated as total number of correct predications divided by 60 (total number of predictions) multiplied by 100.

The resulting predictive accuracies were averaged over the ten replications and the averaged value was taken as the predictive accuracy of the system. The results are presented in Tables 6.1, 6.2, 6.3 and Figure 6.1.

Table 6.3: Predictive accuracy of the system for high support count

Percentage of Patterned data	Predictive accuracy with device assisted mining	Predictive accuracy without device assisted mining
90%	95.00%	95.00%
80%	91.67%	91.67%
70%	80.00%	80.00%
60%	56.67%	55.00%
50%	51.67%	51.67%
Average	75.00%	74.67%

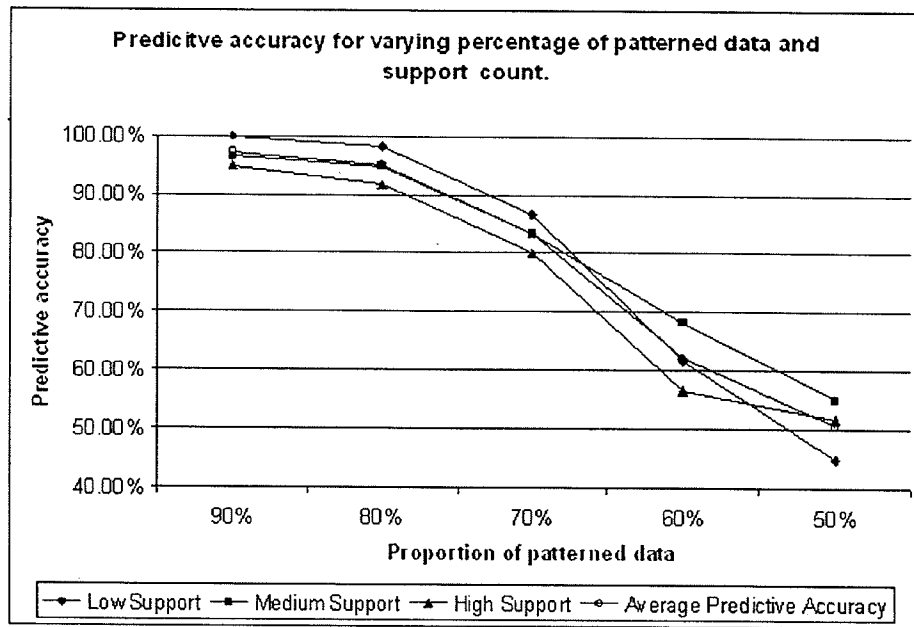


Figure 6.1: Predictive accuracy with device based mining.

6.3 Results and Discussion

The goal of this evaluation was to test the ability of the data miner to predict a user's next action on varying proportions of valid and invalid smart home data. Automating devices wrongly will cause irritation to the user.

From the results presented in Tables 6.1, 6.2, and 6.3, it is inferred that the system performs better in most of the cases when the individual and the day based mining was supplemented with device-based mining for low and medium support counts. This shows that the system can identify overlapping device usage of multiple inhabitants as described in Section 5.4. However when high support count is used to mine the data, device assisted mining is of little use. This can be attributed to the fact that most of the overlapping device interactions do not have the required high support count to be considered as interesting patterns.

The average predictive accuracy of the system with device assisted mining for 90% of patterned data is 97%, for 80% of patterned data is 95%, and for 70% of patterned data is 83% with low, medium and high support counts put together. However, the predicative accuracy deteriorates sharply for lower proportions of patterned data. This can be attributed to the high amount of non-patterned (noisy) data generated.

For higher proportions of valid data the system performs similarly for low, medium and high support counts. However, the system performs better with medium support count for decreasing proportions of patterned data.

When a high support count is used for mining data with decreasing proportions of patterned data, most of the regularly occurring device interactions do not satisfy the support count criteria. Similarly, if a low support count is used for mining with decreasing proportions of valid data, device interactions which are randomly generated satisfy the support count criteria. In both cases the number of wrong predictions increases and the system performs poorly.

It is clear that the system has very good predictive accuracy. Based, on the assumptions, I conclude that our system performance in general is good when the percentage of patterned

data generated is greater than or equal to 70%. The performance of the system is moderate when the percentage of patterned data generated is in the range from 60% to 70%. But our system performs poorly when the percentage of patterned data generated is less than 60%.

This Chapter marks the end of phase 5 of my research.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

In this research we have pursued two comprehensive studies, namely:

- 1) Synthetic data generation for smart homes, and
- 2) Design changes to existing sequential mining algorithms to mine smart home device interaction data for predicting users' next device interactions.

First, we overviewed the existing popular synthetic data generators, identified their limitations and their inability to generate plausible smart home data. Then we outlined our design considerations and methodology, architecture and implementation details for generating plausible smart home data, and discussed our results and limitations. Our two level hierarchical design to generate synthetic data based on spatial scenarios and devices available within that space has helped us to generate plausible and more realistic smart home data. Our data generator has the capability to produce varying proportions of patterned and

non-patterned data based on the input probability transition matrix of the Discrete Time Markov Chain (DTMC). Our data generator uses controlled variation technique to produce repeating device interaction patterns. And our data generator uses different random streams to produce independent replications. All these feature makes our data generator flexible and adaptable.

Second, we overviewed the existing sequential mining algorithms, identified the architectural changes that needed to be done so that they can be used to mine smart home data. We have presented our novel concept of distinguishing between events and significant events. We use the concept of significant events to prune and log the smart home data. Using this concept of significant events, we greatly reduced the amount of useless data that needs to be stored in the database. We mine data based on individual, day of the week, and time of the day as compared to mining data based on a population as in traditional sequential mining algorithms. Mining based on individual user, day of the week, and event duration has helped us to get good predictive accuracies. We presented our design considerations, design methodology, architecture and implementation methodology of our Intelligent Miner and User Action Predictor along with our results and discussions in Chapter 5. Our User Action Predictor, on average, has more than 80% prediction accuracy for input data sets containing more than 70% of patterned data.

7.2 Future Work and Applications

There is a large array of possible future work. In this Section, I briefly discuss some possible future work along with a few indicated in Sections 4.5 and 5.8 of this thesis.

Synthetic Data Generator

At present our data generator is only capable of producing data based on individual users at a particular time; it does not generate data for multiple users at the same time. However, interactions for multiple users can be created by combining the results of multiple runs. In our future work, we can extend the base data generator to support generation of multiple user data. To generate multiple user data we propose to consider independence and non-independence between device interactions of different users. Also, it is our plan to make this data generator client - server based, thereby enabling any external system to communicate their requests to our data generator. This will help to automate the necessary integration with the data mining component of the external systems and avoid the need for manual processing in getting the synthetic data sets to the external data mining system.

Our synthetic data generator produces data repeatedly based on hierarchies and induces time heterogeneity in the generated data. Because of these features our system can be easily altered to produce various proportions of repeating time heterogenous hierarchical data such as, fraudulent and/or non-fraudulent credit card data, telephone bandwidth usage data, and network intrusion data.

Intelligent Miner and the Predictor

Our prototype system is greatly dependent on the server. If the server crashes then there is a danger of all the essential automating services being stalled. However, considering the home gateway's computational capabilities we substantiate the need for using server based services. But in future it would be appropriate to study possible ways to have distributed control and to have failsafe home automation server systems. Also, with the increase in the computational capabilities of home gateways, it should be possible to collect/cache and store some critical data such as mined patterns from server, personal preferences on device automation, etc. on the home gateway and this will substantially reduce the dependence of the gateway device on the server.

Also, since the home gateway needs to be always connected to the Internet in order to communicate with the server, a failure in the Internet connection can lead to stalling of the system. In future, it would be interesting to study the various ways of providing failsafe communication mechanisms between the home gateway and the home automation service provider.

Because of personal data being stored and transmitted over the internet there are privacy issues that needs to be addressed in the future.

At present the mining process is based on support count. However, support-count-based mining can only differentiate between interesting patterns (which have the required support count) and non-interesting patterns (which do not have the required support count) but not between valid (interesting and authentic) and invalid (interesting and non-authentic) data. This is clear from the results presented in table 6.2 i.e., if we use low support count for decreasing proportions of valid data during the mining process, device interactions which are randomly generated satisfy the support count criteria and are considered as interesting patterns by the system. Because of this, the number of wrong predictions made by the system increases. In future, it would be interesting to come up with algorithms which can differentiate patterns based on both their support count and validity.

Our system classifies data based on significant events. In future, it would be interesting to study how the system performs when supplied with regular expression constraints specifying a valid start state and a valid end state of event-sequences. I propose that using regular expression constraints will reduce the number of features to be mined, but however specifying a single regular expression constraint which can represent and/or contain all the possible events would be a complex task. There is also a high chance of missing some important event sequences when using regular expression constraints.

At present our system needs training time. The system collects device interaction data over the period of time and mines the data before automating the home. In future, it should be possible to collect and store the user preference which specify the devices to be automated based on time, user preference, and day of the week. And using this preference

data we can automate the home till our system is sufficiently trained and ready to take over the automation process. We have made a provision for the preference manager in our system architecture, so in future with proper design considerations it should be easy to integrate the preference manager module in our system.

This page is intentionally left blank.

Bibliography

- [1] A. Aboulnaga, J. Naughton, and C. Zhang. Generating Synthetic Complex-structured XML Data. In *Proceeding of the 4th International Workshop on the Web and Databases (WebDB 2001)*, pages 79–84, Santa Barbara, CA, USA, 2001.
- [2] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499, Santiago, Chile, 1994.
- [3] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *Proceedings of the 11th International Conference on Data Engineering (ICDE)*, pages 3–14, Taipei, Taiwan, 1995.
- [4] R. Agrawal and R. Srikant. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proceedings of the Fifth International Conference on Extending Database Technology (EDBT)*, pages 3–17, Avignon, France, 1996.
- [5] S. Ahmed. Design and Implementation of an Extensible Sensing Service for Home Networks. Master’s thesis, Department of Computer Science, University of Manitoba, Winnipeg, MB, June 2006.
- [6] AHRI. Aware Home Research Initiative. <http://www-static.cc.gatech.edu/fce/ahri/>. Last accessed on: June 6, 2006.

- [7] OSGi Alliance. Open Service Gateway Initiative. <http://www.osgi.org>. Last accessed on: June 4, 2006.
- [8] P. Azalov and F. Zlatarova. SDG-A System for Synthetic Data Generation. In *Proceedings of IEEE International Conference on Information Technology: Computers and Communications*, pages 69–75, Washington, DC, USA, 2003.
- [9] D. Bhat. A Framework for Interoperability in Home Networks. Master’s thesis, Department of Computer Science, University of Manitoba, Winnipeg, MB, June 2006.
- [10] S. Butler. Smart Toilets and Wired Refrigerators . *US News and World Report*, 126(22):48–58, 1999.
- [11] Statistics Canada. Number of Persons Aged 80 Years and Over in the Canadian Population, 1956 to 2006 (Table). Portrait of the Canadian Population in 2006, by Age and Sex, 2006 Census. Version updated July 17, 2007 . <http://www12.statcan.ca/english/census06/analysis/agesex/pdf/97-551-XIE2006001.pdf>. Last accessed on: October 2, 2007.
- [12] Statistics Canada. Proportion of Persons Aged 65 Years and Over in the Canadian Population, 1956 to 2006 (Table). Portrait of the Canadian Population in 2006, by Age and Sex, 2006 Census. Version updated July 17, 2007 . <http://www12.statcan.ca/english/census06/analysis/agesex/pdf/97-551-XIE2006001.pdf>. Last accessed on: October 2, 2007.
- [13] Associated Content. Home Automation for the Physically Challenged. http://www.associatedcontent.com/article/4317/home_automation_for_the_physically.html. Last accessed on: October 2, 2007.
- [14] CoSort. Rowgen File Synthesizer, Table Populator, Test Data Builder, Random Data Generator. <http://www.iri.com/public/solutions/rowgen/rowgen.htm>. Last accessed on: January 2, 2007.

- [15] A. Craig. Dundee Presentation. <http://www2.rgu.ac.uk/obj/search/Research/SustainableHousing/Custodian/Home.html>. Last accessed on: October 2, 2007.
- [16] S. K. Das, D. J. Cook, A. Bhattacharya, E. O. Heierman, and T. Y. Lin. The Role of Prediction Algorithms in the MavHome Smart Home Architecture. *IEEE Wireless Communications Special Issue on Smart Homes*, 9(6):7–84, 2002.
- [17] M. H. Dunham. *Data Mining: Introductory and Advanced Concepts*. Pearson Education Inc., Delhi, low price edition, 2003.
- [18] J. M. M. Ferreira, T. Amaral, D. Santos, A. Agiannidis, and M. Edge. The Custodian Tool: Simple Design of Home Automation Systems for People with Special Needs. In *Proceeding of the EIB Scientific Conference*, pages 88–97, New York, NY, USA, 2000.
- [19] M. N. Garofalakis, R. Rastogi, and K. Shim. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB)*, pages 223–234, Edinburgh, Scotland, UK, 1999. Morgan Kaufmann.
- [20] M. N. Garofalakis, R. Rastogi, and K. Shim. Mining Sequential Patterns with Regular Expression Constraints. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):530–552, 2002.
- [21] GSApps. GS DataGenerator—Automated Data Generator: Create Test Data of any Size, Datatype and Complexity. <http://www.gsapps.com/products/datagenerator/index.html>. Last accessed on: January 6, 2007.
- [22] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, first edition, 2001.
- [23] E. Heierman and D. J. Cook. Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns. In *Proceedings of the International Conference on Data Mining*, pages 537–540, Washington, DC, USA, 2003.

- [24] S. Helal, W. Mann, H. El-Zabadani, J. King, J. Kaddoura, and J. Jansen. The Gator Tech Smart House: A Programmable Pervasive Space. *IEEE Computer Journal*, 38(4):50–60, 2005.
- [25] M. Huber. Home Automation for the Physically Challenged. http://ranger.uta.edu/~huber/cse4392_Smarthome/Lectures/Introduction.ppt. Last accessed on: March 6, 2007.
- [26] IBM. alphaWorks : XML Generator : Overview. <http://www.alphaworks.ibm.com/tech/xmlgenerator>. Last accessed on: January 2, 2007.
- [27] Philips Ambient Intelligence. Philips Research—HomeLab. <http://www.research.philips.com/technologies/misc/homelab/index.html>. Last accessed on: June 6, 2006.
- [28] D. Jeske, B. Samadi, P. Lin, C. Rendon, and R. Xiao. Synthetic Data Generation Capabilities for Testing Data Mining Tools. In *IEEE Military Communications Conference (MILCOM 2006)*, pages 537–543, Washington, DC, USA, 2006.
- [29] D. Jeske, B. Samadi, P. Lin, L. Ye, S. Cox, R. Xiao, T. Younglove, M. Ly, D. Holt, and R. Rich. Generation of Synthetic Data Sets for Evaluating the Accuracy of Knowledge Discovery Systems. In *Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD 2005)*, pages 756–762, New York, NY, USA, 2005.
- [30] L. Jiang, D.Y. Liu, and B. Yang. Smart Home Research. In *Proceedings of the 2004 Machine Learning and Cybernetics.*, pages 659–663, Shangai, China, 2004.
- [31] S. W. Kim, M. C. Kim, S. H. Park, Y. K. Jin, and W. S. Choi. Gate Reminder: A Design Case of a Smart Reminder. In *Proceedings of the 2004 conference on Designing interactive systems*, pages 81–90, New York, NY, USA, 2004.
- [32] D. Kleinbrad. Cisco Internet House. http://money.cnn.com/2000/10/06/technology/cisco_house/. Last accessed on: January 2, 2007.

- [33] P. L'Ecuyer, L. Meliani, and J. Vaucher. SSJ A Framework for Stochastic Simulation in Java. In *WSC '02: Proceedings of the 34th conference on Winter simulation*, pages 234–242, San Diego, CA, USA, 2002.
- [34] C. Leung. Advanced Topics in Computer Systems: Data Mining and Data Warehousing Class Notes, March 2006.
- [35] P. Lin, B. Samadi, and D. Jeske. Development of a Synthetic Data Set Generator for Building and Testing Information Discovery Systems. In *Proceedings of IEEE Third International Conference on Information Technology: New Generations (ITNG 2006)*, pages 707–712, Las Vegas, NV, USA.
- [36] E. Lundin, H. Kvarnstrom, and E. Jonsson. Synthesizing Test Data for Fraud Detection Systems. In *Proceeding of the 19th Annual Computer Security Applications Conference (ACSAC 2003)*, pages 384–396, Las Vegas, NV, USA.
- [37] Martel, Laurent, Caron, Malenfant, and ric. Portrait of the Canadian Population in 2006, by Age and Sex, 2006 Census. <http://www12.statcan.ca/english/census06/analysis/agesex/pdf/97-551-XIE2006001.pdf>. Catalogue no.: 97-551-XIE2006001, 2007, Last Accessed October 2, 2007.
- [38] S. Meyer and A. Rakotonirain. A Survey of Research on Context-aware Homes. In *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers*, pages 159–169, Adelaide, Australia, 2003.
- [39] Microsoft. SQL Server 2005 Home. <http://www.microsoft.com/sql/default.mspx>. Last accessed on: March 10, 2007.
- [40] MySQL. The World's Most Popular Open Source Database. <http://www.mysql.com>. Last accessed on: March 2, 2007.
- [41] News@Cisco. Cisco Unveils Internet Home. <http://newsroom.cisco.com/dlls/fspnisapi3934.html>. Last accessed on: January 2, 2007.

- [42] J. Nielsen. *Usability Engineering*. Academic Press, San Diego, CA, USA, 1993.
- [43] D. Norman. *The Invisible Computer*. MIT Press, Cambridge, MA, USA, 1998.
- [44] G. O'driscoll. *The Essential Guide to Home Networking Technologies*. Prentice Hall, Upper Saddle River, 2001.
- [45] Oracle. The World's Largest Enterprise Software Company. <http://www.oracle.com/index.html>. Last accessed on: October 2, 2007.
- [46] S. H. Park, S. H. Won, J. B. Lee, and S. W. Kim. Smart Home Digitally Engineered Domestic Life. *Personal Ubiquitous Computing*, 7(3-4):189–196, 2003.
- [47] H. Pourreza. *Service Delivery And Composition In A Highly Heterogeneous Distributed Environment*. PhD thesis, Department of Computer Science, University of Manitoba, Winnipeg, MB, January 2008.
- [48] S. Rao and D. J. Cook. Predicting Inhabitant Actions Using Action and Task Models with Application to Smart Homes. *International Journal of Artificial Intelligence Tools*, 13(1):81–99, 2004.
- [49] Microsoft Research. Easy Living. <http://research.microsoft.com/easyliving/links>. Last accessed on: June 6, 2006.
- [50] A. Roy, S. K. Das, A. Bhattacharya, K. Basu, and D. J. Cook. Location-aware Resource Management in Smart Homes. In *Proceedings of First IEEE International Conference on Pervasive Computing and Communications*, pages 159–169, Los Alamitos, CA, USA, 2003.
- [51] SqlEdit. DTM Data Generator, Test Data Generator for Database Testing Purposes. <http://www.sqledit.com/dg/index.html>. Last accessed on: January 4, 2007.