# An Analysis of Different Data Base Structures and Management Systems on Clickstream Data Collected from Advocacy Based Marketing Strategies Experiments for Intel and GM

by

Yufei Wang

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 19, 2005 [June 2005]





Author_____
Department of Electrical Engineering and Computer Science
May 19, 2005

Certified by_____
Professor Glen Urban
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

# An Analysis of Different Data Base Structures and Management Systems on Clickstream Data Collected from Advocacy Based Marketing Strategies Experiments for Intel and GM

by

Yufei Wang

Submitted to the Department of Electrical Engineering and Computer Science

May 19, 2005

In Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and Master of Engineering in Electrical Engineering and Computer Science

## *Abstract*

Marketing on the internet is the next big field in marketing research. Clickstream data is a great contribution to analyze the effects of advocacy based marketing strategies. Handling Clickstream data becomes a big issue. This paper will look at the problems caused by Clickstream data from a database perspective and consider several theories to alleviate the difficulties. Applications of modern database optimization techniques will be discussed and this paper will detail the implementation of these techniques for the Intel and GM project.

---

Thesis Supervisor: Glen Urban

Title: David Austin Professor of Marking, Sloan School of Management

**Acknowledgements**

Many thanks to the people around me who made it possible for me to finish my thesis. To my parents for their everlasting love. To Jimmy, Nhu, Goods, Hemond, Drew, Dan, Nuds, Hao, BK, Pwanda, Boob, Bloux, Jo, Di, Elisa, Albs, and anyone else I missed for their support. To Prof. Urban and Stephen Kao for their assistance. To Kelly, for all that she's helped me with.

## Introduction

Scientific studies in the past have all required the analysis of large amounts of data. With the introduction of the computer, the task of data analysis has saved researchers incredible amounts of time and effort. However, just as computers have allowed researchers to get results faster, computers have also increased the limit of data that can be analyzed. Advances in computer hardware technologies have allowed scientists to look at terabytes of data in reasonable time. This increase leads to interesting problems from a computer science perspective.

A big issue that surfaces is space efficiency. The large quantity of data that needs to be considered often creates a problem due to constraints of physical disk storage. Often times, the reason for this is the manner that data is stored. Redundant storage of data creates a big waste in disk space and is prone to incorrect update and deletion methods. Thus, the problems of redundancy become greater when dealing with large quantity of data.

A second issue that arises is the time of data retrieval. As the number of entries increase for any data set, the time that it takes to search and retrieve data will increase. This problem is very serious since many basic data management operations, such as insertion and deletion, are based on data retrieval. Therefore, for any project that involves examination of very big data sets, it is essential to provide some process of returning data quickly.

This paper will look at two real cases where both of the mentioned issues become a problem. Both involve huge databases of data and require fast and efficient solutions to data management problems. This paper will look at some the theories that involve database structures and management and the application of these theories to solve the problems plaguing large data sets.

# Chapter I

# Intel Project

## Intel Project

The Intel project was a collaboration between the MIT Sloan School Center for eBusiness and the Intel Information Technology and Customer Support department. The project conducted research on customer web surfing habits and what factors and influences can be used to develop stronger relations with the customer, specifically building a more trusting relationship. This project used the support and driver download website from Intel's main website and changed not only the text and information, but the visual presentation of the website to gauge the effectiveness of trust building. Changes started with small graphics' placements and became more sophisticated with the additions of personal wizards and audio enhancements. Data was collected through a method called Clickstream, where each clicks of users who visit the support site were recorded, and a comprehensive survey was collected at the end of users' visits to garner customer reactions to the site. The experiment went through several iterations. At every iteration, a specific change was made the site and data was collected to see how the change affected customer reactions. The results helped to gain more insight on how trust can be built on an online environment and how users' behavior can be modeled to predict navigation so more pertinent information and more effective marketing strategies can be deployed. The results of this project are ultimately very valuable tools in online marketing and a better understanding of consumer advocacy that can show how to make sites like Intel's support page to run more effectively and smoothly.

**The Experiment - Setup**

The Intel project was broken down into five stages. At each stage, a specific change would be made to the website to see how this change would affect users' navigation and ultimately how many users were able to successfully find the correct product and download the appropriate drivers. There were two versions of the Intel support page available to users at each stage of the experiment. Users would be randomly selected to view a test version or a control version. The test version of the site would be the experimental site that contains new changes, while the control version would be the previous iteration's test version. This meant that the control site on the first iteration was the original Intel support web site without any changes. The experiment was conducted in this fashion in hopes that the change made at each iteration would improve upon the previous version of the site. The success rate of the site was measured in two ways. One way to measure the success rate of the site was download rate. This measured how many users were able to find a product and download the drivers for that product. By being able to download the drivers for the product, this meant that the user was able to effectively navigate through the site. Higher the download rates translated into a better designed site, which allowed users to find products more easily. The second measure of success for the Intel support site was through a survey taken by users. The survey asked questions regarding trust and how the users felt after using the site. The survey questions helped to gauge how successful the site was able to develop trust and what kind of relationship did the site help to foster with the users. Using a combination of the two measures, it was determined how successful the changes were at each stage.

**The Experiment – Stages**

The first stage of the experiment saw little changes to the actually Intel support site. The purpose of this stage was to validate experimental presentation and data collection methods, and design and create test and control environments. This stage provided validation of the Clickstream data capture method and the randomization of how users were accessing the control and test versions of the site. In addition to validation purpose, new graphics were added to the site to see how these changes can affect users' attitudes. New graphics such as Intel brand logo, the TrustE seal, and the Privacy BBBOnline seal, were added to the site. It was hoped that the addition of these new graphics would improve feelings of trust on the site and allow users to be more confident. After this stage concluded, it was concluded that the Clickstream data capture scheme and the users' randomization process was working according to plan. It was also discovered that the Intel brand seal was very effective and that users perceived the site to address their overall needs. However, the effects of the trust and privacy seal graphics provided little results. There was only a slight increase in the subjective security rating of the site, and it was shown that the seals did not increase the download rates of the site. Although, the seal graphics ultimately provided little help in the creating a more trustful feel for users, this stage was very successful in providing a foundation for future stages of the experiment.

The second stage changed the layout and the wording of the site. The objective of this stage was to increase download completion and success rate; this was done through improving the navigation of the site and provide context sensitive content to the users. The new layout and wording created many positive results. There was a 3.5% increase in successful downloads, a change from 83.5% to 87.0%, a 3.5% increase in trust (overall), and a 4.5% increase in customer satisfaction. Changes like these convert into roughly savings of $11.5 million dollars for Intel per year, this number is calculated based on number of downloads per year times improved success rate times cost saving per call. These initial rudimentary changes to the site shows how a more trust based, easy to navigate website can help on an online environment for any company. These results in the second stage show the importance of this research. The result show how simple changes can expand cost saving, and how that the more innovative changes in the later stages can dramatically increase users' experiences and better cost saving procedures for companies with support websites.

The third stage of the experiment involved a logical wizard, which provided an alternative way for users to search and download products. Originally, the main way that users downloaded from the Intel support site was to use the Intel search engine and through the results recovered from the search engine, download the correct products. The wizard provided another way for users to accomplish this goal. The wizard consisted of a multi-staged, guide where questions were prompted at the user to help determine the product that the user was searching for. The idea behind the wizard was that it was observed that users had a difficult time finding the correct drivers because it was difficult

to find products through the search engine. Users complained about the terrible search process and this was a big roadblock. The wizard's method of asking the users questions was a valuable tool to establish the exact product that the user was looking for. The wizard was easy to navigate and helped the user to come to the appropriate products much faster than the original method. The wizard that was created only worked for the camera products, so not the entire line of Intel products was available. Clickstream and survey data showed a large percentage in success rates as a result of the wizard. Tracking showed that 33% percent of the users chose to the new wizard, and the new wizard caused an increase in success rate of 16.1% versus the control version, which was the site from stage two. Data collected from the surveys showed an increase from 69.8% to 80.0% in download success rates, and Clickstream data showed an increase of 66.0% to 82.1% in successful download rates. These results show a powerful increase in success rates. The ease that the correct drivers were found for the appropriate product compared to the previous procedures was probably the main reason that caused this increase. The following are some comments left on the survey regarding the helpfulness of the new wizard: "I had to use the download advisor because searching for my product gave me too broad results when I wanted just driver update and my product wasn't listed on the menu past PC cams.", "stumbled onto the download advisor after I had done my own search, if that had been more upfront it would have been better". The success of the wizard was a big step in the research, and the next two stages of the experiment will build on the wizard's achievements.

With the success of the wizard, or download advisor, the fourth stage concentrated on how to improve upon the wizard design. It was a central idea of this research project to somehow build a more trusting feel on the site. The wizard was a perfect opportunity to accomplish this. In a sense, the wizard was already a tool that provided personalized interactions with the users. Users were engaged in a question and answer experience, which was comparably high leveled interaction from a web site. The next step would be to create something that would allow the user to build a much deeper and personal relationship with the site. The concept that was finally adopted was to place a picture of a person to the left hand side of the wizard page. The picture would be of a smiling person that would convey a feeling of warmth and delight, please refer to figure 1 to see an example of what the picture looked like and how the site would look like with the addition of the picture. This picture would create a persona so that the user can feel like as if there is a person that is part of the wizard who is personally guiding the user through their search. The picture was also accompanied by contextual instructions and information, so at each step of the wizard the user was always receiving appropriate help. The hope was that by using the wizard with the persona, the users will develop a relationship with the persona and in turn the site. This was shown through a positive increase in results. According to Clickstream data, the wizard helped to increase download rates by 0.3% compared to the control version, this was statistically significant to backup the notion of positive returns of the persona.


*Figure 1.*

Even though positive results were shown with the picture persona version of the site, it was hypothesized that the addition of the persona would somehow have a much bigger impact on the site. The last stage of the experiment focused on how to improve the persona of the site and how to develop a stronger relationship with users. Several ideas were proposed to accomplish this, but the final concept that was implemented was to add an audio portion to the already existing persona. In the last version of the site, the persona was created with a mere picture of a person. To augment to this feeling, audio was added so that at each stage of the wizard, users would receive voice output to help guide them through the wizard, figure 2 illustrates how the site looks with the addition of the audio portion. The audio add-on offered a dynamically expanded experience through client detection to provide better information and guidance to the users. The addition of audio with the picture appeals to users' visual and audio sense, the only two that the internet can convey; so, it was thought that maximizing users' exposure to these sensations will create a longer lasting and stronger relationship. Not only will users be able to navigate the site effortlessly, but users will grow to trust the site and believe in the messages that are on the site. The results were very affirmative in this stage. The wizard

increased success rates by 2.9% percentage against the control version. Also, the combined effect of picture and audio version of the site produced higher download rates than the picture stand-alone version of the site. In total, for this stage, users had a 85.3% percent of success with the wizard was used and 66.0% percent of success when the wizard was not used. This shows the effectiveness of the picture and audio combination and helps to backup value of advocacy and trust based online marketing strategies.


*Figure 2*

**Clickstream data and analysis**

The first part of this experiment focused on site design and ways to develop customer advocacy and trust. Different approaches were used to try to accomplish this using customer survey and Clickstream data to measure the effectiveness of each change. Although Clickstream data was very vital in determining how many users were able to download the correct drivers, which translated directly into site efficiency, the true

potential of the data was not realized. Clickstream data basically provided a record of all pages that every site visitor viewed during the time span that the experiment was running; the level of detail that Clickstream data provided went beyond a simple list of pages, but instead included every picture, java-script, form, product id, and other information for every user. Even though the detailed nature of Clickstream data might seem burdensome and unhelpful at first and that the usefulness of the data has already been extracted from the first part of the experiment, it is exactly the extra details that the Clickstream data provided that can be very pivotal to online marketing research. The second big part of this experiment involves how Clickstream data is used in combination with the Hierarchal-Bayes modeling method to extract, calculate, and predict user paths on web sites. It is also in the second stage where the heavy data analysis of Clickstream data starts and how different database management systems and structures will play a big role in the analysis.

**Potential uses of Clickstream Data**

The internet is currently the fastest growing form of media in recent history. More and more people are using the internet, and the internet's capacity is expanding everyday (DMNews, 2002). People now can use the internet for various reasons ranging from banking to watching movies. With such an explosion of growth, the internet makes for a prime target for marketers. Online marketing has a big potential to reach a full spectrum of people, and with online commerce growing at a fast past, online marketing could become the next big field in marketing.

However, marketing on the internet is rather difficult. Marketers have no information on who is currently accessing a page, so there is no way of personalizing ads or using a specific marketing strategy. This is the biggest problem of online marketing. There is no real way to personalize, so marketers are forced to use blanketing approaches that appeals to everyone. This has limited results and it would be much better if there were some way to deliver better online marketing. There have been simple attempts at personalization on a few online stores. Currently, online stores target visitors (Mena 2001) using many types of information, such as demographic characteristics, purchase history (if any), and how the visitor arrives at the online store (i.e., did the user find the site through a bookmark, search engine, or link on an email promotion). This does have some limited successes, but this experiment proposes that there is a better method in online personalization.

So far, there has been a wealth of information that has been generally unused by marketers, Clickstream data, which records the pages that users view on a website. This source of information has not been used to its full potential due to the lack of methodology to analyze the data (Bucklin et al. 2002). This is a big deterrent due the large amount of information that needs to be processed as will be explained later in this paper. However, the second part of the Intel project will discuss a procedure to analyze Clickstream data using a statistical model on a page by page perspective that will be efficient and create a way for marketers to better personalize online marketing.

The wealth of Clickstream data allows the breaking down of the page records to create page paths for all visitors. A path will be defined as a string of pages that a users views, starting from the first page that the user sees to the last page when the user leaves the site. It is assumed that the user leaves the site if there is a gap of sixty minutes or more between page views. With the creation of paths, analysis can be done of how the users reach the end goal and what of process did the user go through from the first page to the last page. Path data may contain information about a user's goals, knowledge, and interests. The path brings a new facet to predicting consumer behavior that analysts working with scanner data have not considered (Montgomery et al. 2004). Specifically, the path encodes the sequence of events leading up to a purchase, as opposed to looking at the purchase occasion alone. To illustrate this point consider a user who visits the Amazon web site, www.amazon.com. Suppose the user starts at the home page and executes a search for "Liar's Poker", selects the first item in the search list which takes them to a product page with detailed information about the book *Liar's Poker* by Michael Lewis (1990). Alternatively, another user arrives at the home page, goes to the business category, surfs through a number of book descriptions, repeatedly backing up and reviewing pages, until finally viewing the same *Liar's Poker* product page. (Montgomery et al. 2004).

Which user is more likely to purchase a book: the first or second? Intuition would suggest that the directed search and the lack of information review (e.g., selecting the back button) by the first user indicates an experienced user with a distinct purchase goal. The meandering path of the second user suggests a user who had no specific goal and is

unlikely to purchase, but was simply surfing or foraging for information (Pirolli and Card 1999). It would appear that a user's path can inform about a user's goals and potentially predict future actions (Montgomery et al. 2004).

The second part of the Intel proposes to use a statistical model that can make probabilistic assessments about future paths including whether the user will make a purchase or in the case of Intel support site, a successful download. The results show that the first user in the example above is more likely to purchase. Moreover, the model can be applied generally to predict any path through the web site. For example, which user is more likely to view another product page or leave the web site entirely within the next five clicks? Potentially this model could be used for web site design or setting marketing mix variables. For example, after confirming that the user will not reach a certain end page, the site could dynamically change the design of the site by adding links to helpful pages, while for those users likely to purchase the site could become more streamlined. A simulation study using the model suggests that download rates on the support site could be improved using the prediction of the model, which could substantially increase operating profits (Montgomery et al. 2004).

**The Model – Background and Hierarchal Bayes**

The ability to personalize marketing using users' paths is a powerful tool. After a few clicks, marketers will be able to tell with high probability whether the user is in a browsing or goal oriented mind state, and provide the correct information and site format

that will create the highest success rates. In order to produce a model that can predict the complex behaviors such as users' click paths, there is a large number of parameters and variables that need to be taken account of. Plus, the heavy size of the Clickstream data, around 250 gigabytes, makes this task even harder to analyze. The statistical model used in this process is a special field under Bayesian data analysis. The idea behind the model is that given data of results of a procedure, in this case the Clickstream data that allows us to see which pages the users have been viewing, Bayesian analysis will allow researchers to work backwards to construct a model that we can use to reproduce clicking behaviors for all future users. Given the large number of parameters that need to be approximated, a special field of Bayesian analysis fits very well into the study. Hierarchical Bayes has two levels. At the higher level, we assume that individuals' parameters (betas or part worths) are described by a multivariate normal distribution. Such a distribution is characterized by a vector of means and a matrix of covariances. At the lower level we assume that, given an individual's betas, his/her probabilities of achieving some outcome (choosing products, or rating brands in a certain way) is governed by a particular model, such as a linear regression (Gelman et al. 2003). This matches the users' clicks behavior model very well.

**The Model - Description**

The model that we developed is a utility based model that is very similar to the one used in the *Modeling Online Browsing and Path Analysis Using Clickstream Data by*

Montgomery et. al (2004) paper. We will first introduce the model and then discuss each of the components.

$U_{pti} = \sum \Phi_{pc} U_{c(t-1)i} + \gamma_{pi} X + \varepsilon$

p – this denotes a specific page
t – time
i – user

This model builds upon the utility model used in the Montgomery et. al paper where the utility of viewing a future page depends on the page the user is currently viewing plus the characteristics of the page. In the form that is written above, the model specifies how the utility of a page p for a user i at time t is equal to the summation of the utility for the same user at t-1 modified by an autoregressive factor to simulate forgetfulness plus a gamma parameter variable that captures the effects discussed later and a final error term.

*Utility from the previous time period*

The summation represents the utility the user obtained from the previous time period. We are considering two possible definitions for the autoregressive $\Phi$ term: $\Phi = \Phi^{(1+\beta\tau)}$ and $\Phi = e^{\Phi \log(1+\beta\tau)}$; both definitions express the time delay effect of the user forgetting more as more time goes by. $\tau$ is the time that the user took to transition from one page to another, and $\beta$ is a parameter that will estimate to see how much $\tau$ will effect the forgetfulness of the user.

*Page characteristics and the Test/Control and Monthly effects*

Outside of utility from a previous time period, there are other factors that need to be modeled. First, there was a significant difference between download rates of the test site and the control site. The Intel project was set up so that the test site of the previous phase became the control site in the next phase in the experiment. One should expect that the download rates should be same when the switch happened since it is still the same site. However, there was a difference in the download rates. Hardware factors can play a role; since the test site and the control site are hosted different servers, so, there might have been a hardware issues that affected download or navigation.

Second, since we have data regarding when a user access a page on the download site, we can use this information to gauge how much time a user takes to go from one page to the next can affect the page path. We decided to use this information on the auto-regressive part of the model, as we will assume that the longer a user takes in between page visits the more the user will forget, thus having more of an effect on the latent utility.

Third, download rates were increasing as the experiment progressed. This could have been due to users in general becoming more accustomed to the internet and were able to navigate site with more ease. This too will be incorporated into the model to see how progression in calendar time will affect user page paths.

All these factors are incorporated into the $\gamma_{pi}$ X part of the model. We will define the $\gamma_{pi}$ variable as the following: $\gamma_{pi} = \gamma^{B}_{pi} + \gamma^{T}_{p}$(Test) + $\gamma^{M}_{p}$ f(calendar time). This will try to capture the Test/Control effect and the Monthly effect, which are described above, in the model. $\gamma^{B}_{pi}$ is a base variable where we hope to include the rudimentary demographics information that we have about the users, such as speed, browser type, etc. We define $\gamma^{B}_{pi} = \Gamma_p\, d_i$, where $\Gamma_p$ is a parameter and $d_i$ is a vector of user characteristics, which we can obtain from the database. $\gamma^{T}_{p}$(Test) will encapsulate the Test/Control site effect. $\gamma^{T}_{p}$ will be a parameter while (Test) will have a value of 0 or 1 depending on whether the page was on the Test site or the Control, this information can also be retrieved from the database. $\gamma^{M}_{p}$ f(calendar time) will denote the monthly effect. f(calendar time) will be a lookup function which will associate a value for all the calendar times that the project was in running. This way, the parameter variable $\gamma^{M}_{p}$ will be able to estimate the effect of the progression of the time on user download rates.

The X vector will represent the current page that the user is viewing and also the user's choice of wizards. Part of the Intel Project involved wizards, which were implemented on the download website to help users to better find products. There were three wizards in total available to users: simple, pictures, and audio. Users have the option of selecting any one or none. We first define a vector that has four elements; we'll call this vector the wiz vector. The first element is defaulted to having the value of one, and the rest of the elements will represent the user's choice of wizards with 0 by not using and 1 meaning selection. An example of a wiz vector would look like this. If the user choose to use the simple and picture wizard but not the audio wizard, then the wiz

vector would like [1,1,1,0]. The X vector then is defined using wiz vectors. X will be a vector with 4*C elements formed by concatenating C wiz vectors. All but one of the wiz vectors will be zero and the one nonzero wiz vector will correspond to the page that the user is currently viewing. For example, if there are three categories of pages, and setting the wizard choices the same as before, then if the user is currently viewing category two the X vector would look like the following: [0,0,0,0,1,1,1,0,0,0,0,0].

Finally there is an error term, which we set for now to be a normal with mean of zero and sigma of one at the end of the model. The exact equation of the model is always changing. New factors and variables will added and subtracted as the analysis changes, but, the next step of the process to input the correct data. The data that will be used with this is the Clickstream data, however, it is in its raw state and needs to be parsed, cleaned, and paths need to be formed to correctly do the analysis. The rest of paper will explain how all of this process will be done.

**The Data – Clickstream**

Clickstream data was very important in the first part of the Intel project. Clickstream data gave results to show how many users were able to successfully download drivers at every stage of the experiment. In this stage, Clickstream data plays a much bigger role. Clickstream data will be analyzed to put the pages that users view in order to create paths. These paths will be closely examined to try to capture users' behavior and characteristics. Correctly and quickly handling the massive Clickstream

data will be very important in this part of the Intel project and will remain the focus for the rest of this paper.

Clickstream data may be unfamiliar to some readers so this section will explain the nature and format of Clickstream. The Clickstream data that is used in this project is very simple in concept. Intel web-server computers that contain and serve all the pages for its support site can log all the pages that users request to see. These logs simply turn into Clickstream data. The logs record any URL viewed by the user in their browser window, along with other information. Since it records the actual pages viewed in the browser window, it avoid the problems of caching that is commonly found with Internet Service Providers (ISP). The only problem with this form of recording is that the way that users reach the website is lost. For example, it will be unclear if the user came from a bookmark or if the user has two browser windows open and is switching between windows. This will be problematic in later parts of the analysis.

Intel completely provided all the Clickstream data and with no data parsing at all. Therefore, the data was received was in the format that was recorded by Intel's web servers. This is rather confusing and it took some time to completely understand all the information that was passed by Intel. The following will clearly describe the format and values of all the parameters that passed in the Clickstream data. The raw log data files contain lines of records ordered by time such that each line represents a http quest from a user. Each line is in the format of:

#Fields: date time c-ip cs-username s-sitename s-computername s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status sc-win32-stat us sc-bytes cs-bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referer)

Here is an example of an entry in the raw log file:

*2002-07-01 00:00:52 10.9.130.25 - W3SVC2 ARISTOTLE02 10.9.142.233 80 GET /scripts-df/filter_results.asp strOSs=44&strTypes=DRV%2CUTL&ProductID=180&OSFullName=Windows*+XP+Professional&submit=Go%21 200 0 0 64 3 2 97 HTTP/1.1Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)DF%5Fredir=NoRedir;+surveyUID=102548232 9776A589162;+surveysite=1;+surveyBreak=1;+dev_ro=1http://downloadfinder.intel.com/sc ripts-df/Product_Filter.asp?ProductID=180*

This table will explain what each variable is and how it can be useful in the analysis.

| | |
|---|---|
| Date | Date that the user made the http (web page) request |
| Time | Time that the user made the http (web page) request |
| c-ip | IP of the user |
| cs-username | not used |
| s-sitename | not used |
| s-computername | Name of the server computer |
| s-ip | IP of the server |
| s-port | Port that the request came through |
| cs-method | not used |
| cs-uri-stem | Page requested by the user (page that the user is viewing) |
| cs-uri-query | not used |
| sc-status | not used |
| sc-win32-stat | not used |
| Us | not used |
| sc-bytes | not used |
| cs-bytes | not used |
| Time-taken | not used |
| cs-version | OS of the user |
| cs(User-Agent) | Type of web browser that the user is using |
| cs(Cookie) | Cookie of the user |
| cs(Referer) | Page from which the user made the page request |

The table shows the abundance of information that is captured in the Clickstream data. This is good from a modeling perspective since it will provide information on many variables about the user and the user's actions. However, this can be terrible from an analysis perspective, since it will be very expensive time and computation wise to parse

all the Clickstream data into a correct format and then storing it into a storage saving and effective way. The rest of the paper will explain how this was done and go into detail on how databases play a big role in handling and storing the large Clickstream dataset.

**The Data - Parsing**

The files provided by Intel are in server log format, so, the files need to be parsed and put into a suitable format for analysis. The files are in plain text files; therefore, an excellent choice to parse the data is to use Perl. Perl is an interpreted scripting programming language that is known for its power and flexibility. It combines the familiar syntax of C, C++, sed, awk, grep, sh, and csh into a tool that that is more powerful than the separate pieces used together. In order to completely parse all of the Clickstream data, several Perl scripts are needed to produce the correct data. This section will go into more depth on each of the scripts and how they work together.

There are many entries in the Clickstream data that do not pertain to this study, such as requests for gif files or css (style sheet) files. This is where strip.pl (the .pl extension just denotes that the strip file is a Perl script) comes in. This perl script will go through the raw log files and delete all entries of user http requests that wouldn't make sense for the project. These entries are generally user requests for picture files and other html format files and would not provide any information to the pages that users are viewing. Here is the full list of conditions to determine which raw log entries that are taken out.

.gif/
.jpg

.css/
.js
/scripts-util/
/T8Clearance/
/camera_top.asp/
/camera_nav.asp/
/camera_left.asp/
/camera_id.asp/
/OS_top.asp/
/OS_nav.asp/
/OS_left.asp/
/OS_content.asp/
/Detail_top.asp/
/Detail_nav.asp/
/Detail_left.asp/
/Detail_content.
/audcamera_top.asp/
/audcamera_nav.asp/
/audcamera_left.asp/
/audcamera_id.asp/
/audOS_top.asp/
/audOS_nav.asp/
/audOS_left.asp/
/audOS_content.asp/
/audDetail_top.asp/
/audDetail_nav.asp/
/audDetail_left.asp/
/audDeta
il_content.asp/

Any user http requests in the Clickstream data that contains the extensions above will be deleted by strip.pl. After strip.pl is run, reduced versions of the raw log files are created with the same name with a .reduced extension. Running this script greatly reduced the amount of data that needs to be analyzed.

Now that the Clickstream data has been reduced, two processes will start. Sample1 to sample4 is run for one process and survey1 and survey2 is run for a different one. All these scripts function in about the same fashion. The purpose of both of these processes is to sample from the entries of the reduced Clickstream data. Sample1 to sample4 will conduct more of a random sampling process, while survey1 and survey2 samples only entries of users that have filled out the visitor surveys. These sampling

processes are necessary from a statistical point of view to create a good data set for data analysis.

*Sample1.pl to Sample4.pl*

| | |
|---|---|
| Sample1.pl | This script will go through a reduced raw log file looking for surveyUID, a unique user identifier for each user. If a surveyUID exist in the entry, then it will output the product id that exists in the entry to a text file. If the product id is a /camera.asp or /graphic.asp the product id is outputted to -1. Note the product id will be output as one of the following: 596, 594, 459, 351, 595, 425, 628, 352, 350, 460, 354, 353, 356, 355 or else it will be outputted as 0. This text file is called surveyuids.$type.txt |
| Sample2.pl | Counts the category of product ids. -1 is wizard, non zero is camera, and 0 is everything else |
| Sample3.pl | This script will go through all the users and determine if the user viewed the wizard or was looking at camera products. If the user did view one of these, they will be classified as a camera user or a wizard user. If the user does not fall into these two categories, then with 1% probability they will be selected. |
| Sample4.pl | All users that have viewed the wizard or looked at camera products have been selected. Also, 1% of the population has been randomly selected. This script will go through the entire reduced Clickstream data set and select any entry that associated with any of the selected users. |

*Survey 1 to Survey 2*

| | |
|---|---|
| Survey1 | This script will extract all surveyIDs from 1804il-final.txt and 1393il-final.txt, which I assume to be raw survey data. I took a look at these two files and they do indeed seem to be raw survey files. All survey ids are outputted to a file called surveyuids.survey.txt. |
| Survey2 | This script will go through all the survey ids that were created with the Survey1 script and go through each reduced file to create a sample file. Each sample file will contain all the entries from the raw logs that contain the surveyID's found by the Survey1 script. |

Regardless of whether the Sample 1 to 4 scripts or the Survey 1 to 2 scripts are run, both process should produce a file with the sample extension. This file should contain all entries in the Clickstream data associated with users that are selected dependent on the process that was used. With this sample file, it is now possible to get the salient information, paths information, for all the selected users. For this part of data processing, much of the work will involve database interactions. The following will describe which scripts are used to create paths and other user information, more detail will be given about the database tables and organization later on.

*Paths, Products, Users, and Visits*

| | |
|---|---|
| Path1 | First part of the paths creation procedure. This script will sift through the |

sample file and extract fields such as time, date, user-id, page that is viewed, agent, and status_id. After collecting this information, the script connects to the database and inserts a row into the paths table.

Path2

This is the second part of the paths creation process. The path1 script went through the sample file and gamers the appropriate information from each entry in the Clickstream data. Path2 will first create a list of all unique users. Then, it will go through each user and select all the pages that the user views sorted by time. With this sorted list, the script will incrementally sift through each page examining the time between each page viewing. If the time between page-viewings is greater than 60 minutes, it is determined that a path has stopped and a new one has started. This process creates all the paths for each user, and after the paths have been resolved, they are inserted into the database along with other information such as visit number, number of pages in the paths, and so on.

Visits

This script will compute all the needed parameters of the visits table. Similar to the Path2 script it will first obtain a list of unique users and then go through each user to gain the information. First it will count the number of visits that for each user and record the maximum number. Then, it will calculate the duration of each visit with the starting time and the ending time. Finally, the script will acquire some exiting information and insert all these information into the database.

Products

The product scripts are very similar to the paths scripts. This is the first part of the procedure. Again a list of unique users is first made, then for each

user the products script queries the existing paths table for all the pages that the user views. From this list of pages, the script will look for any product-ids. The product ids are embedded in the page url so the task of extracting product ids is not too difficult. When ids are found, the script makes an insertion into the database with the correct information.

Users

This is a very simple script that populates the users table in the database with the appropriate information such as user id, control or test, how many visits does this user make to the site, and so on.

Products2

The reference table of products is created with this script. Basically the reference product table lists what product id is linked to what product. This script reads this information from a reference file and creates the table in the database.

Visits2

Determines the is_camera_wizard, is_graphics_wizard, is_camera_product for the visit table. **Note:** is_camera_wizard is determined if the users visits /camera.asp page, is_graphics_wizard is determined if the user visits /graphics.asp page, and is_camera_product is determined if the product id of one of the pages is the following: 350, 351, 352, 353, 354, 355, 356, 425, 459, 460, 594, 595, 596, 628

This describes all the scripts that are used to parse and format the raw Clickstream data. The scripts are depended the order that they are run, so anyone repeating this analysis should run the scripts in the correct order. The scripts are presented in the order that they should be run, this is just a reminder:

1) strip.pl

For random sampling:
2A) sample1.pl
2B) sample2.pl
2C) sample3.pl
2D) sample4.pl

For sampling of only survey users:
2A) survey1.pl
2B) survey2.pl

3) path1.pl
4) path2.pl
5) visits.pl
6) products.pl
7) users.pl
8) products2.pl
9) visits2.pl

## Database and Database design

In the Clickstream data, a random sampling of one percent of the total user population will result in about forty to fifty thousand users. Assuming on average that each user will view a minimum of ten pages, this means running one random sampling will require the storage of at least four hundred to five hundred thousand pages. In addition to this, storage must be provided for user information, visits, product ids, paths, and survey data. The task of handling storage for this large amount of information is very difficult. An efficient scheme is needed to well organize the data, make it so that the data can be accessed computationally fast and correct, take up the least amount of physical disk space. The rest of this paper will explain how this was done for the processed Clickstream data. These sections will describe what theories are behind the database management structures and the exact implementations.

## Data Models

Storing a large amount of data is ultimately a very difficult job. Challenges include answering questions about the data quickly, allowing changes from different users to be made consistently, and allowing access to certain parts of the data correctly. The DataBase Management System (DBMS) is a piece of software designed to make those tasks easier. By storing data in a DBMS rather than as a collection of operating system files, we can use the DBMS's features to manage the data in a robust and efficient manner. As the volume of data grow, DBMS support becomes indispensable.

The user of a DBMS is ultimately concerned with some real-world enterprise; the data to be stored describes various aspects of this enterprise. In the Intel project, there are pages, users, and other information, and the data in the Clickstream database will describes these entities and their relationships. A data model is a collection of high-level data description constructs that hide many low-level storage details. A DBMS allows a user to define the data to be stored in terms of a data model. There are two main data models: the relational data model and the semantic data model.

**Semantic Data Model**

While the data model of the DBMS hides many details, it is nonetheless closer to how the DMBS stores data than to how a user thinks about the underlying enterprise. A semantic data model is a more abstract, high-level data model that makes it easier for a user to come up with a good initial description of the data. These models contain a wide

variety of constructs that help describe a real application scenario. A DBMS is not intended to support all these constructs directly; it is typically built around a data model with just a few basic constructs, such as the relation model. A database design in terms of a semantic model serves as a useful starting point and is subsequently translated into a database design in terms of the data model the DBMS actually supports.

A widely used semantic data model is called the entity-relationship (ER) model. The ER model allows the data involved to be described in terms of objects and their relationships and is widely used to develop an initial database design. It provides useful concepts that facilitate the change from an informal description of what users want from their database to a more detailed, precise description that can be implemented in a DBMS. The ER model concentrates first on requirement analysis. This step is usually used to find out what users want from the database. This is a very informal step that understands what data is to be stored in the database, what applications must be built on top of it, and what operation are most frequent and subject to performance requirements. The second step is the conceptual database design phase. This again is an information gathering stage used to develop a high-level description of the data to be stored in the database, along with the constraints known to hold over this data. The goal is to create a simple description of the data that closely matches how users and developers think of the data (and the people and processes to be represented in the data). This is the main point of the ER model. The last step is the logical database design stage. This stage is used to choose a DBMS to implement the database design, and convert the conceptual database

design into a database schema in the data model of the chosen DBMS. The task in the logical design step is to convert the ER schema into a relational database schema.

**Relational Data Models**

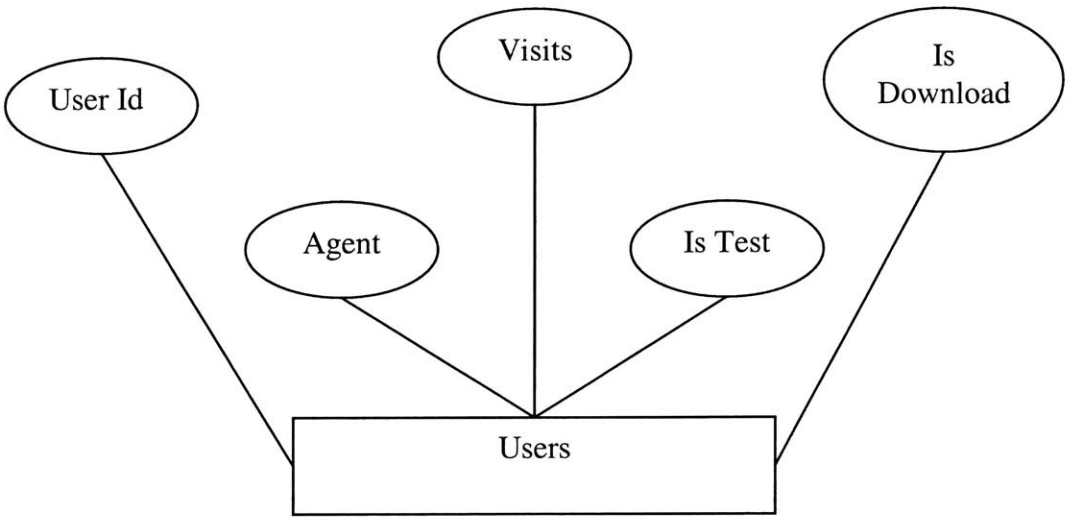
The central data description construct in this model is a relation, which can be thought of as a set of records. In the relational model, the schema for a relation specifies its name, the name of each field (or attribute or column), and the type of each field. For example, movies in a movie rental store database may be stored in a relation with the following schema: Movies( mid: *string*, title: *string*, genre: *string*, year: *date*). The schema says that each record in the Movies relation has four fields, with field names and types as indicated. An example instance of the Movies relation appears below.

| mid | Title | Genre | Year |
|---|---|---|---|
| 53666 | Star Wars: Episode III | Sci-Fi | 2005 |
| 53688 | Kingdom of Heaven | Action | 2005 |
| 53650 | Sin City | Action | 2005 |
| 53831 | Wedding Crashers | Comedy | 2005 |

Each row in the Movies relation is a record that describes a movie. The description is not complete, for example the actors and actress in the movie are not included, but is presumably adequate for the intended applications in the rental store's database. Every row follows the schema of the Movies relation. The schema can therefore be regarded as a template for describing a movie.

**Logical Database Design: ER to Relational**

The ER model is convenient for representing an initial, high-level database design. Given an ER diagram describing a database, a standard approach is taken to generating a relational database schema that closely approximates the ER design. Below is the users ER diagram.




User entities in a tabular format:

| user_id | is_test | visits | agent | is_download |
|---|---|---|---|---|
| 1000030304140A880369 | T | 1 | Mozilla/4.0 | f |
| 1000061395280A267354 | F | 1 | Mozilla/4.0 | T |
| 1000076525100A238123 | F | 1 | Mozilla/4.0 | T |
| 1000110037141A805555 | F | 1 | Mozilla/4.0 | T |
| 1000132080970A235880 | F | 1 | Mozilla/4.0 | T |

Without a lack of constraints, the mapping of the Users relation is very straightforward. Each attribute of the ER becomes an attribute of the table. The following SQL statement captures the proceeding information.

```
CREATE TABLE "users" (
    "user_id" varchar(20) NOT NULL,
    "is_test" boolean NOT NULL,
    "visits" smallint NOT NULL,
    "is_download" boolean NOT NULL,
    "agent" varchar(5000) NOT NULL,
    UNIQUE(user_id)
```

```
    );
```

With the same kind of procedure to similar ER diagrams of the rest of the relations in the Clickstream data, the following SQL commands creates the tables that are needed to establish all the relations in Clickstream.

```
CREATE TABLE "paths" (
    "path_id" integer DEFAULT nextval ('path_id_seq') NOT NULL,
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint,
    "page_number" smallint,
    "url" varchar(5000) NOT NULL,
    "parameters" varchar(5000),
    "agent" varchar(5000),
    "page_time" decimal(5,2),
    "timestamp" timestamp NOT NULL,
    "is_test" boolean NOT NULL,
    "load_balance_id" smallint,
    "status_id" smallint,
    UNIQUE(user_id,timestamp,url)
    );
CREATE INDEX paths_path_id ON paths(path_id);


CREATE TABLE "products" (
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint NOT NULL,
    "product_id" smallint NOT NULL,
    UNIQUE(user_id,visit_number,product_id)
    );
CREATE INDEX products_product_id ON products(product_id);


CREATE TABLE "products2" (
    "product_id" smallint NOT NULL,
    "name" varchar(500),
    "status" varchar(50),
    UNIQUE(product_id)
    );


CREATE TABLE "visits" (
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint NOT NULL,
    "start_time" timestamp NOT NULL,
    "end_time" timestamp NOT NULL,
    "visit_time" decimal(5,2) NOT NULL,
    "exit_page" varchar(250) NOT NULL,
    "page_count" smallint,
    "is_test" boolean NOT NULL,
    "is_camera" boolean DEFAULT 'f' NOT NULL,
    "is_graphics" boolean DEFAULT 'f' NOT NULL,
    "is_camera_product" boolean DEFAULT 'f' NOT NULL,
    UNIQUE(user_id,visit_number)
    );
CREATE INDEX visits_exit_page ON visits(exit_page);
CREATE INDEX visits_is_camera_wizard ON visits(is_camera_wizard);
```

```
CREATE INDEX visits_is_graphics_wizard ON visits(is_graphics_wizard);
CREATE INDEX visits_is_camera_product ON visits(is_camera_product);


CREATE TABLE "paths" (
    "path_id" integer DEFAULT nextval ('path_id_seq') NOT NULL,
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint,
    "page_number" smallint,
    "url" varchar(5000) NOT NULL,
    "parameters" varchar(5000),
    "agent" varchar(5000),
    "page_time" decimal(5,2),
    "timestamp" timestamp NOT NULL,
    "is_test" boolean NOT NULL,
    "load_balance_id" smallint,
    "status_id" smallint,
    UNIQUE(user_id,timestamp,url)
    );
CREATE INDEX paths_path_id ON paths(path_id);


CREATE TABLE "products" (
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint NOT NULL,
    "product_id" smallint NOT NULL,
    UNIQUE(user_id,visit_number,product_id)
     );
CREATE INDEX products_product_id ON products(product_id);


CREATE TABLE "products2" (
    "product_id" smallint NOT NULL,
    "name" varchar(500),
    "status" varchar(50),
    UNIQUE(product_id)
     );


CREATE TABLE "visits" (
    "user_id" varchar(20) NOT NULL,
    "visit_number" smallint NOT NULL,
    "start_time" timestamp NOT NULL,
    "end_time" timestamp NOT NULL,
    "visit_time" decimal(5,2) NOT NULL,
    "exit_page" varchar(250) NOT NULL,
    "page_count" smallint,
    "is_test" boolean NOT NULL,
    "is_camera" boolean DEFAULT 'f' NOT NULL,
    "is_graphics" boolean DEFAULT 'f' NOT NULL,
    "is_camera_product" boolean DEFAULT 'f' NOT NULL,
    UNIQUE(user_id,visit_number)
     );
CREATE INDEX visits_exit_page ON visits(exit_page);
CREATE INDEX visits_is_camera_wizard ON visits(is_camera_wizard);
CREATE INDEX visits_is_graphics_wizard ON visits(is_graphics_wizard);
CREATE INDEX visits_is_camera_product ON visits(is_camera_product);
```

## Schema Refinement and Normal Forms

ER models and relational data models give a good starting for the final database design. The SQL commands above create tables that consider the relations between the data variables. Now, it is time to consider performance criteria and typical workload. The following sections will use the relation schemas created from before and refine them using constraints. It will start with an overview of the schema refinement approach, and then introduce the main class of constraints used in refinement, functional dependencies. After that, "normal forms" of databases will be looked at and how normal forms can be accomplished through decomposing databases to reduce redundancy.

**Redundancy**

Storing the same information redundantly, that is, in more than one place within a database is a very big issue in database design. Redundant storage of information is the root cause of many problems that can lead to serious performance and physical disk space issues. Here are the four major problems caused by redundancy:

- redundant storage – some information is stored repeatedly
- update anomalies – if one copy of such repeated data is updated, an inconsistency is created unless all copies are similarly updated
- insertion anomalies – it may not be possible to store certain information unless some other, unrelated, information is stored as well
- deletion anomalies – it may not be possible to delete certain information without losing some other, unrelated, information as well.

An example of redundancy and the problems caused by redundancy is as follows. Consider a relation on the hours worked by employees of a company. Hourly_Emps(ssn, name, lot, rating, hourly_wages, hours_worked). The key for the relation is ssn. In addition, the hourly_wages attribute is determined by the rating attribute. That is, for a given rating value, there is only one permissible hour_wages value. A table of values for this example could be this.

| Ssn | name | lot | Rating | hourly_wages | Hours_worked |
|---|---|---|---|---|---|
| 123-22-2666 | Chen | 48 | 8 | 10 | 40 |
| 213-32-5332 | Chueng | 22 | 8 | 10 | 30 |
| 235-23-5432 | Hu | 35 | 5 | 7 | 30 |
| 848-67-1299 | Lin | 35 | 5 | 7 | 32 |
| 938-55-8388 | Shu | 35 | 8 | 10 | 40 |

If the same value appears in the rating column of two entries, one constraint tells us that the same value must appear in the hourly_wages column as well. This redundancy has the same negative consequences as the ones just listed.

- Redundant Storage: The rating value 8 corresponds to the hourly wage 10, and this association is repeated three times.
- Update Anomalies: The hourly_wages in the first entry could be updated without making a similar change in the second entry.
- Insertion Anomalies: We cannot insert an entry for an employee unless we know the hourly wage for the employee's rating value.

- Deletion Anomalies: If we delete all entries with a given rating value (e.g., we delete the entries for Hu and Lin) we lose the association between that rating value and its hourly_wage value.

Ideally, we want schemas that do not permit redundancy, but at the very least we want to be able to identify schema that do allow redundancy. Even if we choose to accept a schema with some of these drawbacks, perhaps owing to performance considerations, we want to make an informed decision. Note: It is worth considering whether the user of null values can address some of these problems. As we will see in the example, we can deal with the insertion anomaly by inserting an employee entry with null values in the hourly wage field. However, null values cannot address all insertion anomalies. For example, we cannot record the hourly wage for a rating unless there is an employee with that rating, because we cannot store a null value in the ssn field, which is a primary key field.

**Functional Dependencies**

A functional dependency, FD, is a kind of constraint that generalizes the concept of a key. Let R be a relation schema and let X and Y be nonempty sets of attributes in R. We say that an instance r of R satisfies the FD $X \rightarrow Y$ if the following holds for every pair of entries $e_1$ and $e_2$ in r: if $e_1.X = e_2.X$, then $e_1.Y = e_2.Y$. We use the notation e1.X to refer to the projection of entry $e_1$ onto the attributes in X. An FD $X \rightarrow Y$ essentially says that if two entries agree on the values in attributes X, they must also agree on values in

attributes Y. It is important to note that an FD is not the same as a key constraint. Functional dependencies merely point out the relations that exist in database. There are a few rules that functional dependencies follow that we can use later for refining our database schema:

- Augmentation: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z.
- Transitivity: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$
- Union: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
- Decomposition: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$

**Normal Forms**

The next important concept in database refinement is normal form. Given a relation schema, we need to decide whether it is a good design or we need to decompose it into smaller relations. Such a decision must be guided by an understanding of what problems, if any, arise from the current schema. To provide such guidance, several normal forms have been proposed. If a relation schema is in one of these normal forms, we know that certain kinds of problems cannot arise. The challenge of database enhancement is fit the database into a normal form.

While studying normal forms, it is important to appreciate the role played by functional dependencies, FDs. Consider a relation schema R with attributes ABC. In the absence of any constraints, any set of entries is a legal instance and there is no potential for redundancy. On the other hand, suppose that we have the FD $A \rightarrow B$. Now if several

entries have the same A value, they must also have the same B value. This potential redundancy can be predicted using the FD information. If more detailed constraints are specified, we may be able to detect more subtle redundancies as well.

The normal forms based on FDs are first normal form (1NF), second normal form (2NF), third normal form (3NF), and Boyce-Codd normal form (BCNF). These forms have increasing restrictive requirements: Every relation in BCNF is also in 3NF, every relation in 3NF is also in 2NF, and every relation in 2NF is in 1NF. A relation is in first normal form if every field contains only atomic values, that is, no lists or sets. This requirement is implicit in our definition of the relational model. Although some of the newer database systems are relaxing this requirement, in this chapter we assume that always holds. 2NF is mainly of historical interest. 3NF and BCNF are important from a database design standpoint.

*Boyce-Codd Normal Form and Third Normal Form*

Let R be a relation schema, F be the set of FDs given to hold over R, X be a subset of the attributes of R, and A be an attribute of R. R is in Boyce-Codd normal form if, for every FD $X \rightarrow A$ in F, one of the following statements is true:

- $A \varepsilon X$; that is, it is a trivial FD, or
- X is a superkey.

Intuitively, in a BCNF relation, the only nontrivial dependencies are those in which a key determines some attributes. Therefore, each entry can be thought of as an entity or

relationship, identified by a key and described by the remaining attributes. BCNF ensures that no redundancy can be detected using FD information alone. It is thus the most desirable normal, from the point of view of redundancy.

The definition of 3NF is similar to that of BCNF, with the only difference being the addition of a rule. Let R be a relation schema, F be the set of FDs given to hold over R, X be a subset of the attributes of R, and A be an attribute of R, R is in third normal form if, for every FD $X \rightarrow A$ in F, one of the following statements is true:

- $A \varepsilon X$; that is, it is a trivial FD, or
- X is a superkey.
- A is part of some key for R

To understand the third condition, recall that a key for a relation is a minimal set of attributes that uniquely determines all other attributes. A must be part of a key (any key, if there are several). It is not enough for A to be part of a superkey, because the latter condition is satisfied by every attribute. Finding all keys of a relation schema is known to be an NP-complete problem, and so is the problem of determining whether a relation schema is in 3NF.

**Decomposition**

Intuitively, redundancy arises when a relational schema forces an association between attributes that is not natural. Functional dependencies (and, for that matter, other constraints) can be used to identify such situations and suggest refinements to the

schema. The essential idea is that many problems arising from redundancy can be addressed by replacing a relation with a collection of "smaller" relations. A decomposition of a relation schema R consists of replacing the relation schema by two, or more, relation schemas that each contain a subset of the attributes of R and together include all attributes in R. Intuitively, we want to store the information in any given instance of R by sorting projections of the instance. The goal of decomposition is to reduce any relation to normal forms. If a relation schema is in one of these normal forms, we know that certain kinds of problems cannot arise. Considering the normal form of a given relation schema can help us to decide whether or not to decompose it further. Therefore, the goal of any decomposition procedure is to get rid of redundancy in a relational schema and reduce the schema into a normal form. The following is a widely accept algorithm to decompose a relation given a set of functional dependencies. (Tsou, Fischer 1982).

Input : A set U of attributes and a set F of FD's over U .
Output : A decomposition of U with a lossless join, such that every set in the decomposition is in BCNF under F .
Procedure :
begin
1. X + U ;
2. $Y \leftarrow \emptyset$
** after initialization, we start the main loop * *
3. while X $\neq$ Y do
begin
4. Y $\leftarrow$ X
5. Z $\leftarrow \emptyset$ ;
6. repeat : if IYI>2 then do
begin
7. for each attribute A $\varepsilon$ Y do
8. for each attribute B $\varepsilon$ Y-A do
begin
** check if Y-AB $\rightarrow$ A is logically implied by F * *
9. if A $\varepsilon$ CL(Y-AB) then d o
begin
** remove *B* from Y * *
10. Y $\leftarrow$ Y-B ;
11. Z $\leftarrow$ A ;
12. go to repeat ;
end ;
end ;
end ;
** remove Z from X and output a relation scheme Y * *

```
13. X ← X-Z
14. write Y
end ;
end ;
```

## Decomposition of the Intel Clickstream Data Set

In previous parts, the process that Clickstream data is parsed and filtered for the needs of this project is illustrated. It is evident that an effective storage schema is needed based on the size of the data. From the theory and the concepts that have been introduced in the segments above, the following sections will develop a working database of tables in normal form that will include all the information need for the Hierarchal Bayes analysis.

From the already documented Perl scripts, this is a list of all the variables in the Clickstream data that can be garnered: path_id, user_id, visit_number, page_number, url, parameters, agent, page_time, timestamp, is_test, load_balance_id, status_id, is_download, start_time, end_time, exit_page, is_camera_wizard, is_graphics_wizard, is_camera_product, product_id, name. All these variables correspond to a unique path of pages viewed by a unique user with a set of characteristic variables that denote the nature of the path. Since the Hierarchal Bayes analysis is only concerned about paths, it is adequate to use one large table to storage all the data. However, we will see that this will introduce many problems. Here is an example of some issues that can be encountered:

- Every time a new url is inserted into the database, the same product_id, is_camera_wizard, etc. values must be entered as well. If there is a mistake in the

entry of the extra characteristic values, the database would contain different information on the same urls. This is also a storage waste, since the extra information has to be inserted into the database every time a user views the url.

- Every time a product is view by a user, the same information that describes the product has to be inserted into the database. If there is a mistake in entering the information, there would be a discrepancy in the database. Again, this is an indication of redundancy.

**Functional Dependencies and Tables**

It is clear that the option of storing all the Clickstream data in one large table is not wise. Data is stored redundantly making errors easy to propagate, and the task of retrieving information will face heavy performance burdens because of the large size. Therefore, it is necessary to decompose the large table into much more efficient datasets. The first step in doing this would be to determine the functional dependencies of the data. From the url and paths centered nature of the Clickstream data, the FDs generally are also centered around paths and urls. Here is the list of FDs that describes all the relations in the Clickstream data.

- user_id, timestamp, url → path_id, visit_number, parameters, agent, page_time, is_test, load_balance_id, status_id
- user_id → is_test, visits, is_download, agent
- user_id, visit_number → start_time, end_time, visit_time, exit_page, page_count, is_test, is_camera_wizard, is_graphics_wizard, is_camera_product

- user_id → visit_number, product_id
- product_id → name, status

With a full list of FDs, it is now possible to complete the decomposition. Following the algorithm that is presented before, a set of smaller, more proficient tables is constructed:

```
        Table "public.products"
  Column   |       Type        | Modifiers
-----------+-------------------+-----------
user_id     | character varying(20) | not null
visit_number | smallint           | not null
product_id  | smallint           | not null

        Table "public.products2"
  Column  |       Type        | Modifiers
----------+-------------------+-----------
product_id | smallint           | not null
name      | character varying(500) |
status    | character varying(50)  |

              Table "public.visits"
    Column      |          Type          | Modifiers
----------------+------------------------+-----------
user_id         | character varying(20)  | not null
visit_number    | smallint               | not null
start_time      | timestamp without time zone | not null
end_time        | timestamp without time zone | not null
visit_time      | numeric(5,2)           | not null
exit_page       | character varying(250) | not null
page_count      | smallint               |
is_test         | boolean                | not null
is_camera_wizard  | boolean              | default 'f
is_graphics_wizard | boolean             | default 'f
is_camera_product | boolean              | default 'f

          Table "public.users"
  Column   |       Type        | Modifiers
-----------+-------------------+-----------
user_id    | character varying(20) | not null
is_test    | boolean            | not null
visits     | smallint           | not null
is_download | boolean           | not null
agent      | character varying(5000) | not null

                      Table "public.paths"
   Column    |          Type          |              Modifiers
-------------+------------------------+-------------------------------------------
path_id      | integer                | not null default nextval('path_id_seq'::text)
user_id      | character varying(20)  | not null
visit_number | smallint               |
page_number  | smallint               |
```

```
url           | character varying(5000)    | not null
parameters    | character varying(5000)    |
agent         | character varying(5000)    |
page_time     | numeric(5,2)               |
timestamp     | timestamp without time zone | not null
is_test       | boolean                    | not null
load_balance_id | smallint                 |
status_id     | smallint                   |
```

The creation and use of these tables show the application of modern database refinement techniques. Staring with ER models, and then using functional dependencies to decompose datasets into normal form, this method of creating a resourceful database is highly efficient and is very useful in eliminating redundancy. The database that is created through this method will handle large workloads and use little physical storage space as possible. With such a helpful database, completing the Hierarchal Bayes portion of the Intel project will be much more smooth and efficient.

# Chapter II

# GM MyAutoAdvocate Project

## Background

With the growing popularity of the Internet, consumers nowadays can easily access the product information needed to compare all available options that best suit their demands. With the spread of e-commerce, consumers no longer need to rely on local stores and agencies to provide for their needs. The simplified transactions that the Internet enabled have made it easy for consumers to switch between different venders or service providers regardless of their physical locations (Urban, Oct. 2003). All these factors are leading to an empowered consumer who has gained an edge on marketers.

With this shift in power, businesses have felt the pressure to switch to other methods to draw new customers and keep existing customers loyal. One such method is trust-based marketing. An emerging theory, trust-based marketing is a concept that has been championed by Prof. Glen Urban of the Sloan School of Management at MIT. He believes that consumers will be drawn to companies that they trust. Furthermore, they will reward companies that are always open and complete with information by trusting information presented by the company.

This section looks into how marketing has changed as a result of the Internet, focusing heavily on trust based marketing. Next it discusses how the GM-MIT project is studying trust-based marketing. Finally, it discusses the individual trust components that were implemented in the My Auto Advocate web site.

***Trust***

"Trust is the key in an advocacy strategy." (Urban, Oct. 2003) The trust dimension, a concept by Prof. Urban, presents different levels of trust customers have in a business. Depending on the level of trust in a company, consumers will have different attitudes towards a company. The purpose of any trust-based marketing campaign is to develop the right level of trust to keep customers and bring in new ones. As Urban preaches, the key to marketing in an era of consumer power is to be open and honest to inspire trust with the consumer. This concept has not always been prevalent, even in today's marketing campaigns, but the future may find the majority of companies turning to this strategy. This section discusses how trust has changed in the different phases of marketing: past, present and future.

**Past: Push-Based Marketing**

Pure push marketing practice, which involves virtually no trust, exists at one extreme of the trust dimension. Push based marketing is an aggressive promotion of a product. The advertisement is one sided and tends to use limited information (deception) and tricks to draw users to a product. In a push based business model, a company benefits in the short run by taking advantage of the imbalance of information and capitalizing on consumers that are not yet fully informed about the product features and services. The goal is to get as many sales as possible, especially sales of high-margin items, through the use of flashy media advertising, aggressive promotions, and one-sided communication that may mislead customers. (Urban, Oct. 2003)

Because companies are faster then consumers at responding to new forms of information, companies in the past have always had an edge. Hence, push-based marketing has been the core of marketing techniques for the past 50 years.

**Present: An Assault on Traditional Push-Based Methods**

The Internet has changed the imbalance of information between consumers and marketers. Companies cannot assume that consumers lack knowledge on certain product features and services. Because of this, push-based marketing techniques have started to flop. Companies who mislead consumers suffer significant loss of trust and this effect is long term and extremely detrimental. Any short-term sales gain that companies earn from a push-based marketing campaign cannot make up for the loss in future sales.

Companies have started to realize that they need to build a partnership with the consumer. This technique is known as relationship marketing. Backed by Customer Relationship Management and one-to-one marketing concepts, a partial trust-based company targets consumers better and are more efficient in delivering persuasive information and personalized promotions. In this manner, marketing techniques revolve around informing and building a partnership with consumers. Companies have started to move away from assuming that the consumer knows nothing and must be "pushed" a product. Moving along the trust spectrum, they have moved towards higher trust and increased advocacy.

In today's marketing campaigns, Customer Relationship Management strategies are just emerging. In the next few years, however, this strategy will grow increasingly popular to accommodate the empowered consumer. Although push-based marketing techniques will continue to dominate for the next few years, the Internet will slowly set in motion a strong migration towards trust building.

**Future: A Move towards Trust-Based Marketing**

Recent marketing techniques that revolve around Customer Relationship Management is a result of the new consumer: one who feels he or she is knowledgeable enough to make the right purchasing decision with the right information. But eventually, relationship marketing will lose its effectiveness due to the ever continuing shift in consumer power. "People are more educated and informed than ever, and they have the tools to verify a company's claims and seek out superior alternatives from competitors." (Urban, 2004)

Consumer advocacy, a full trust-based strategy, exists at the opposite end of the trust dimension from the traditional push based marketing strategy. A full trust-based business aims to build long lasting relationships with its customers by truly representing the customers' interests. A company that adopts this strategy may then do something that puts a consumers interest (the cheapest product that satisfies its requirements) in from of the company's interest (having a consumer by its product). This may mean recommending a competitor's product. By doing this, the company builds a strong trust

relationship with the consumer where the consumer can count on the company to look after its needs.

Consumer Advocacy is the epitome of a trust-based marketing which also encompasses building a relationship with a consumer and focusing on customer satisfaction. Advocacy goes beyond being a trusted advisor to the consumer. An advocate goes out of the way to proactively represent the customer's interests. For example, the company will attempt to contact the necessary government agency if the customer has problems with another company. The GM project aims to eventually move the My Auto Advocate concept towards complete customer advocacy. Below is an advocacy pyramid which exemplifies these concepts. Note that the top of this pyramid is customer advocacy; advocacy can only work if all other parts of the pyramid are functioning smoothly.

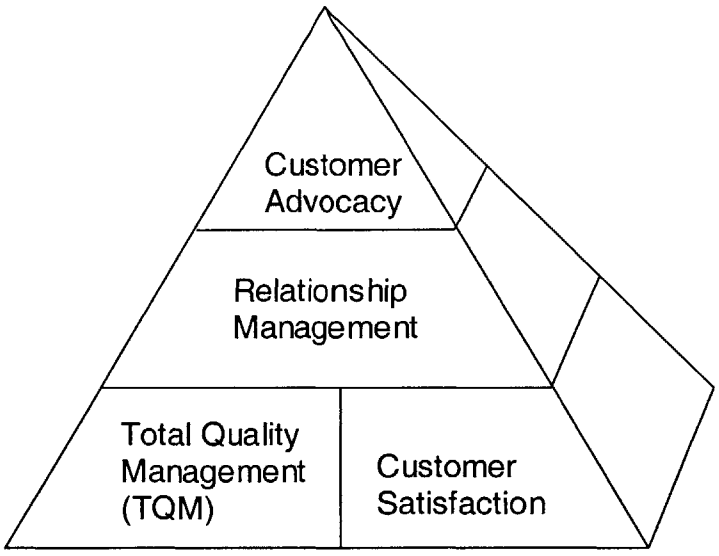



**Figure 1: Customer Advocacy Pyramid**

**The Effect on Automobile Manufacturers**

Automobile Manufacturers have been notorious for their push-based marketing techniques. From their ad campaign to their pressure sales approach at a car dealership, auto manufacturers do not have a high trust factor with consumers. General Motors is no exception. It spent more then 5 billion dollars last year on advertising. Much of their advertising campaign has focused on aggressively promoting their cars through glitzy, slick ads, one of the cornerstones of traditional push-based marketing.

Traditional push-based marketing ads are losing their effectiveness. Consumers have started to realize that educating themselves on automobiles is much more important then relying on a TV advertisement. The Internet has enabled this new reliance on information. Automobile specifications and competitive price comparisons are a large part of the information explosion on the Internet that has benefited consumers. Through sites such as Amazon and epinions, consumers can find other consumer's opinions on makes and models. Through online pricing sites, consumers can find comparisons with other vehicles and also find out how much a model is selling for. Price incentives and other push-based marketing techniques cannot work as effectively because consumers can use the Internet to determine whether the final sticker price is actually competitive. Many other advantages that manufacturers have had in the past have also been diminished due to the Internet. Hence, manufacturers such as GM have needed to look to alternative methods to distinguish themselves from each other.

### *GM Advocacy Project*

GM has worked with Prof. Urban and the E-business at Sloan group for many years. The past few years have produced the seeds for many of the novel marketing projects currently being employed by GM. These include the Auto Show in Motion experience and the Auto Choice Advisor website. However, GM as a whole still has a long way to go in terms of advocacy. The majority of GM marketing techniques still revolve around push-based marketing. Much of these techniques work well for a short time period and then quickly lose any positive impact that they had. As a rule, most of these techniques hurt the company and its image in the long run.

The goal for the joint project with MIT Sloan is to change the culture of push-based marketing. Most consumers are wise to many of the push-based marketing techniques. Moreover, many of GM's current techniques reinforce negative stereotypes on GM cars and further push down GM consideration.

### GM's Need for an Advocacy Based Site

This past fiscal year, GM announced a 5.45 billion dollar loss. This is a substantial loss that has lead many to challenge the leadership of GM. In addition, there has been a strong call for changing the way GM does its business. MIT's project fits in with these goals. This project is attempting to show the effectiveness of different trust-based techniques to encourage GM to change the way it does business.

Although GM has made strides towards building some level of trust with the customer, GM leadership still seems entrenched in old techniques. A common tactic used this past year by GM was to increase the MSRP while simultaneously offering huge discounts. While this tactic had a short positive impact on sales, the long run has been very negative as it only reinforced the idea that GM cars are cheaply made. In addition, consumers came to believe that GM cars were only worth what they were after the discounts. This hurts GM in multiple ways. In addition to losing trust in customers, it also hurts GM's ability to market vehicles at a reasonable MSRP.

Because of their past difficulties, an advocacy approach is strongly needed. GM needs to repair its image and inspire trust among car buyers. GM can do this by working on their Customer Relations Management. They need to work to build a relationship with the consumer from the beginning of the buying phase to throughout the ownership of a GM vehicle. GM has worked hard through the owner center (https://www.mygmlink.com) to build the relationship with the consumer after the vehicle has been bought. They have used this project as an engine towards helping bridge the gap between consumer and GM during the buying experience.

**MyAutoAdvocate**

The MyAutoAdvocate project fits into GM's need for advocacy very well. The heart of the MyAutoAdvocate project is advocacy and the project's sole purpose is to empower the customer with knowledge through a fashion that is believable and trusting.

MyAutoAdvocate will provide not only raw information but give the user a complete experience. The following will describe the parts that make up MyAutoAdvocate and other key components and how each of these parts helps to create and develop trust with customers.

**Opt-In Ads**

One of the biggest ideas of advocacy is non-pushy advertisement. As described in the background section, a tenet of advocacy is to develop trust with the customer and move away from the push based advertising. This became an issue for attracting users to the site. Since the majority of advertising strategies in use up to now are push based, the MyAutoAdvocate project needed to create a way to draw users to the site but still adhere to the trust style of the site. This is how the opt-in ads were created. With knowledge empowerment and trust as the themes, ads were set up in strategic car web sites. The ads stressed knowledge and tried to attract users by showing the incredible amount of information that they can obtain from the site. The ads assert that informed consumers make better purchase decisions and pulls at users' feelings to be smart consumers. This allows the users to be in a mind set of trust and knowledge seeking as they come to the MyAutoAdvocate site. Users already start off knowing that the MyAutoAdvocate site there to help them and provide them with all the information they need to make smart vehicle buying choices.

**Community – Communispace**

A strong sense of community has always been regarded as a key ingredient in building trust amongst a group of people. From the early stages of the MyAutoAdvocate site, it was felt that the site needed to build an easily accessible community to the users. Initially, it was difficult to conceive this idea within an online environment; but, the idea of Communispace, an online forum, was soon found and adopted. The online forum is a website where users can come to view, contribute, or start new threads of discussion. Communispace is an implementation of the online forum with cars and GM products as the central themes of discussion. Users can also come to Communispace to express views about the MyAutoAdvocate site and to meet other users. In addition to feedback, the main goal of Communispace is relationship and community building. With this in mind, Communispace is setup to create an environment where communication is safe, easy, and very fluid.

Communispace was originally not part of any MIT research. Communispace was first used mostly to serve as a focus group and as a way for GM to test out new ideas. GM had been using Communispace as a venue to get opinions on possible future endeavors on hybrid and hydrogen fuel cars. The MyAutoAdvocate project presented a great opportunity for Communispace. Communispace grew from serving a focus group and panel function to providing a gathering place for users to congregate and exchange opinions. This allowed the size of Communispace to really grow, and Communispace now functions as a key part of the MyAutoAdvocate project.

One of the main attractive points of Communispace is the lack of censorship on all materials that are discussed. Users are free to talk about any subject without the fear of some authority suppressing the discussion even if the material is damaging to GM. This is a very novel idea. Since the MyAutoAdvocate site is sponsored by GM, it would be reasonable for the makers of the site to remove any content that does not paint GM in the best light. However, this is not the case. Users are encouraged to express their views, however negative or positive of GM vehicles, without any fears of censorship. This way, users liberally and freely express ideas, and users know that the other users on Communispace are also doing the same. With no censorship, Communispace puts an incredible amount of trust in the users. By showing such faith, users are much more likely to reflect this trust. Allowing negative comments on Communispace has another added value as well. Even if negative comments can cause harm to GM's image, users know that comments on Communispace are sincere. When positive comments are made, these comments are much more likely to be believed. Users are generally cynical and distrustful of any positive contributions on the forums, because it can be perceived as pushy advertisements. Also, since GM is sponsoring the site, users could think that only people who think highly of GM cars frequent the forum; therefore, there could be the possibility that there is some inherent positive bias to Communispace. By permitting negative comments, users know that all comments made on Communispace are authentic. Therefore, when users make good comments about GM cars, other users are much more likely to believe it. Users will know that the comments are coming from another user who genuinely believes in them. This will allow other users to accept the comments much easier and allow the persuasive powers of the comments to be much more effective.

This underlines the use of trust in advertising and how creating trust with customers should be a focal goal for marketers.

Another advantage of using Communispace is the forum's property of drawing back users through self governance. The format of Communispace is such that at first, threads of discussion will be moderated, but as time goes on, Communispace will become completely self governing. Self moderation expands upon the no censorship doctrine of Communispace to truly give users the freedom to explore any topic. This freedom encourages discussion and allows users themselves to exercise discretion on the topics and materials that are talked about. This way, a true sense of a trusting community is built using the free flow of information and peer to peer judgment of the appropriateness of the material as the foundations.

**VASIM (Virtual Auto Show In Motion)**

The Virtual Auto Show In Motion (VASIM) or GM Proving Grounds is the central premise of the MyAutoAdvocate site. The objective of the VASIM is to reproduce the experience of attending an Auto Show In Motion (ASIM) but on an online setting. The reasoning for this is due to the success of the ASIM event. The ASIM is tailored to people who are considering purchasing a vehicle, but, are not the kind of people that usually attend auto shows. At the ASIM, visitors can take an up close look at different cars and can even get a chance to drive some of the vehicles. Extensive information is readily available, and attendees are encouraged to mingle with each other

to offer advice and get first hand experience from other people. The ASIM is particularly beneficial for GM. At the ASIM, attendees are exposed to not only GM vehicles but cars from all major car manufactures. Visitors can personally compare different cars, and, it is through this first-hand comparison that users can really see the difference between GM vehicles and how GM cars are superior to its competitors. The ASIM is a great event for showing the outstanding quality of GM cars and this is the biggest reason to try to bring this experience to as many people as possible using VASIM. To recreate the experience of ASIM, users that go through the VASIM should be exposed to all the things that the ASIM offers. The ASIM is a live event, which lets users to walk around and go through many informational talks, take test drives, and listen to user testimonials. The hope of VASIM is to allow users to be fully submerged in the experience and appeal to the users' senses so strongly that users feel as if they are physically there at the ASIM.

This feeling is created through the GM Proving Grounds. Through a clever interweaving of technology and media, the GM Proving Grounds, which is the formal name for the VASIM, provides users with the unique experience of ASIM in a virtual setting. Everything is done to try to build the excitement and energy of the real ASIM event. The Proving Grounds puts users in the driver seat of a car through rich movies of live test drives and allows users to watch exciting user testimonials of ASIM attendees. Visitors of the Proving Grounds site receive all the information and benefits of attending the real ASIM, and it is hoped that the Proving Grounds with produce similar results as to the ASIM and allow GM's commitment to top-notch quality shine through.

**ASIM - Auto Show in Motion**

The success of the ASIM is also a big influence for the choice of how the MyAutoAdvocate site is setup. The MyAutoAdvocate site should have a more interesting way to present information to users than the traditional html arrangement of frames or menus. The idea is to somehow immerse the users with graphics and media to recreate a feeling of attending a real auto show. Given the technologies at the time of the implementation of MyAutoAdvocate, Apple's VR technology's ability to construct a moving 3-D image was perceived to be the best choice to use as the main navigation tool. The 3-D image would be of an autoshow; so, users would immediately get the feelings of an auto show when they arrive at the MyAutoAdvocate site. Users would be able to use the mouse to change the orientation and perspective of the image. This would allow the user to move the mouse to "look around". This 3-D image approach of presenting the site is called the Virtual Auto Show or VAS. Users then would be able to click on certain parts of the image to get further information on a specific topic. For example, users would click on the booth with the OnStar logo and would be redirect to the official OnStar web site. Or, users can click on an image of a group of people and be redirect to Communispace. Please see the diagram below for an illustration how the VAS could look like.

The Apple VR technology was very promising; however, given the timetable for implementing the MyAutoAdvocate site, it was decided try an easier technology. Instead, the Macromedia Flash design was picked. Flash could create the same effects that the Apple VR; but, the scroll of the image would be reduced to just left and right, where the Apple VR allowed 360 degrees of change. Also, Flash would reduce the quality of the image that is viewed; so, the overall experience for the user would be reduced. This was thought to be acceptable since the time that was given was limited and Flash was still capable of delivering a wonderful experience and providing the needs of a navigation tool. The diagram below shows the how the VAS is finally produced with Flash.


**ACA**

The Auto Choice Advisor (ACA) and car specifications portions of the MyAutoAdvocate site provide the majority of the content for users. People who come to MyAutoAdvocate are usually interested in cars and most are considering buying a vehicle in the near future; therefore, car specifications and ACA are great tools for users to become familiar with different vehicles and find cars that are suited for them. ACA is a large database that contains detailed information on vehicles on all major car manufactures. The idea behind ACA was a research project done at MIT from 1997 to 2000 called Trucktown. Trucktown was a research project that provided users with an online advisor that guided them through the selection process of buying a new pickup truck. The success of the Trucktown project prompted GM to implement the project for all GM vehicles. This was how ACA was created; although, ACA now consists of all manufactured vehicles instead of just GM cars. ACA still retains the same advisory setup

as Trucktown, but the scope of choices and information is on a much bigger scale. ACA starts with users answering a few questions regarding the background of the user, personal preferences, price range, and what the user is looking for in a car. Then, based upon the answers that the user has given, ACA will provide the user with several cars that ACA believes are a good match. These matches are placed in a "virtual garage" so the user can do an even more in-depth comparison of the results. The initial questions allow users to narrow down choices to a small number, and the "virtual garage" lets users do a through comparison to come to a concrete decision. The user then can use ACA's extensive database to lookup more information or find dealers and vendors. Even though ACA is a GM funded research project, the results are completely unbiased and do not favor any model or car manufacture. The cars that are returned to the user are simply the best match for the user's needs. This, again, is the manifestation of the advocacy and trust theme of the MyAutoAdvocate site.

**Car Specifications**

MyAutoAdvocate also offers detailed specifications on cars for almost all major brands and manufactures. Offering this kind of information has a very positive effect; since all the visitors of the site show strong interest in cars, car specifications give the users a more in-depth look at cars and allow users to make much better comparisons. All kinds of information ranging from front row passenger space to engine torque are available to users. The car specifications that are offered on the MyAutoAdvocate site are also presented in a similar fashion as ACA. No preference is given and only unbiased

facts are presented to the users. This format of facts reporting creates an open environment that just reiterates the advocacy based theme of the site.

**Brochures**

Similar to offering car specifications, brochures from different car manufactures are also available. This again is to help add to the feel of an auto show where information such as brochures would be readily available to anyone. Because of legal reasons, the MyAutoAdvocate site does not actually host have any the brochures but provide links to brochures from other web sites. The brochures mostly come in Adobe Acrobat format and are usually scans of real brochures obtainable in dealerships. These brochures are a good way for users to become more familiar with vehicles that they are interested in and offer a good opportunity for users to explore new cars as well.

**Research Link, Helpful Advice, and Advisors**

In addition to the already mentioned resources, MyAutoAdvocate also offers other information to visitors. There is a page of research links and helpful advice which offers a myriad of sources for assistance on cars and car purchasing. Information vary from government agencies to affordable financing and insurance options. Users can go to this page to explore more about certain options and to be exposed to opportunities that may have not been considered. These helpful tips are there to support users in any car related questions and let the users know that the MyAutoAdvocate site is there to help

them. In terms of advocacy, these research links are present to echo the belief that the MyAutoAdvocate site exists to solely benefit users. MyAutoAdvocate provides the information to make sure that consumers make the decisions that are the best informed and most comfortable with.

## Rewards

The idea of rewards was first considered as a way to attract users to sign up for the Auto Show In Motion events. Since the event was a live event that needed a large number of people to be successful, the use of rewards would help to attract more people to consider going to the event. Users would be awarded a certain amount of points for signing up for ASIM, later; the users can redeem the points for Amazon coupons for chances to enter into a sweepstakes drawing. Using rewards was a great idea and help to contribute the larger turnout to the ASIM. The idea of rewards expanded as time went by. It was realized that it would be cost effective to offer rewards for all sections of the MyAutoAdvocate site. This made it so there would not be any misconceived conceptions of rewards, since before it was only offered to ASIM. Also, this gave people that would normally only concentrate on one part of the site more of an incentive to explore all the different parts and receive a more complete experience.

## Data Control and Database Refinement

Once the MyAutoAdvocate web site is implemented and goes live, the biggest concern for the project becomes data control and database management. With about five thousand users accessing the site, MyAutoAdvocate needs to track all the pages that the users view and the reward points that all the users receive and redeem. This again becomes a very complex problem that is very similar to the situation with the Clickstream data for the Intel project. We will use the same concepts that were used in the Intel

project to refine the data storage for MyAutoAdvocate and create a database that is well managed and organized. This will make analysis of data much easier and improve the performance of data retrieval.

**Data Needs of MyAutoAdvocate**

With a significant number of users visiting the MyAutoAdvocate web site everyday, there is a large amount of data that needs to be stored. First, users' rewards points need to be tracked and monitored to ensure that users are getting the correct points for visiting different treatments. Then, information about the users such as login and contact information must be held as well. Plus, all the users' actions on the sites need to be recorded as well. All the pages and all the different parts of MyAutoAdvocate that the users visit need to be documented. These storage requirements place a lot of conditions and constraints on the database for all the data.

An incredible amount of data needs to be stored for the functions of the MyAutoAdvocate site. First, with about five thousand users, the MyAutoAdvocate site needs to keep track of five thousand user accounts. Each of the accounts contain basic information such as name, email, address, gender, age along with background data like operating system, modem speed, the browser that the user is using, and date that the user created the account. Also, detailed information about when the user joined the site, which survey period, and what was the method that the user arrived at the site are all

recorded. This information is mostly saved for research purpose later on when heavy data analysis is performed; but, it is also preserved for the sake of the users who might forget their passwords or need things shipped their addresses. Rewards and reward point redemption will require attention as well. Users will receive a certain amount of reward points for attending specific events or visiting special portions of MyAutoAdvocate and these rewards points need to be tracked. The accuracy of rewards tracking is very important; since it is essential to maintain a healthy level of trust with the users, there can not be any mistakes with giving rewards to users when they deserve it. Once users receive reward points, the points can be used to redeem Amazon coupons or enter into the cash sweepstakes. Again, it is important to track the number of points that each user has used to redeem for which reward program.

Analogous to the Clickstream data in the Intel project, MyAutoAdvocate also tracks all the pages that the users view. Initially, this is a good way to gauge the effectiveness of the different parts of the MyAutoAdvocate site. It was easy to tell which parts of the site were popular by simply looking at the traffic. It was important to get fast responses on how many people were signing up for the site and what content the users were looking at. The basics of tracking the users include recording the pages that the users viewed and the activities that the users participated in. This involves documenting which urls the users see and the number of points that the users gain from each activity. In addition to this, other interesting data is also collected. For example, the online brochures that are available on MyAutoAdvocate are monitored. Every time a user views a brochure, which brochure was viewed and by whom is recorded. Car specifications

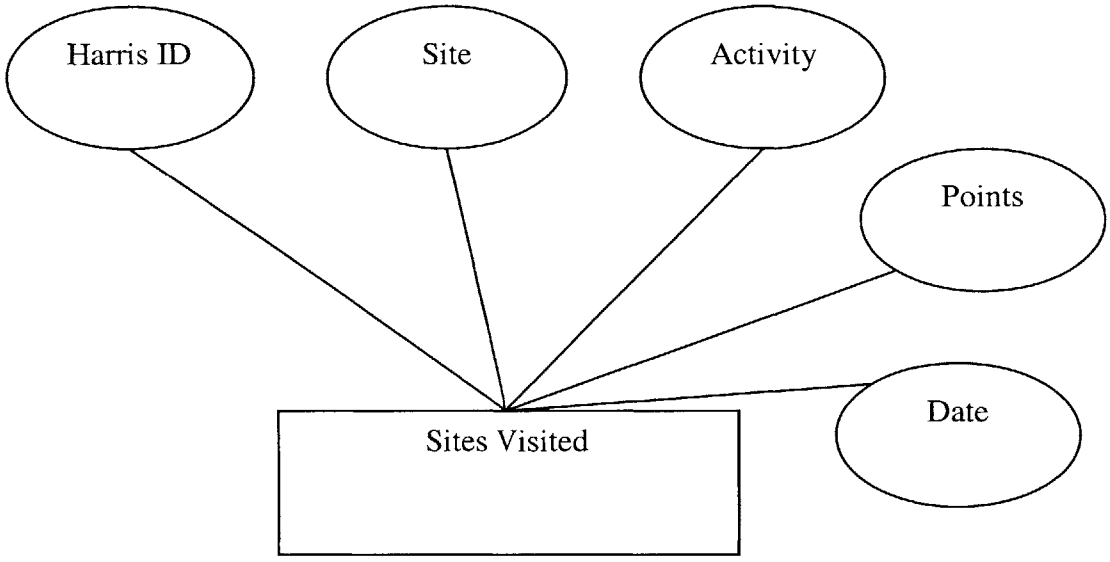
function in a similar fashion. Whenever a user requests to see the specifications of a vehicle, all relevant information is recorded. Plus, even simple clicks are recorded. The helpful tips and research part of MyAutoAdvocate contains links to many other sites that contain useful information. When a user clicks on one of these links, again pertinent information like when the click occurred and the user's id are all recorded. All this data will be very useful later on in the data analysis part of the project. All the data will give researchers freedom to look for certain patterns or behaviors that might normally not be captured. This will be especially helpful in the Hierarchical Bayes portion of the project to look for personalization of the user behaviors.

**Initial Database Refinement – Data Models**

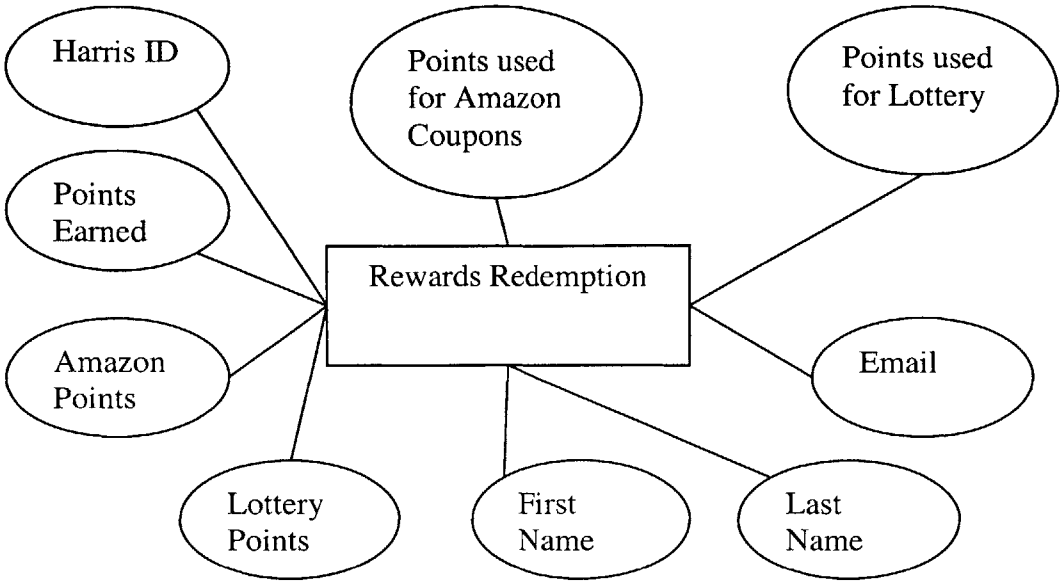
The process used to optimize the database for the MyAutoAdvocate site will be very similar to the methodology used in the Intel project. First, we will look at the MyAutoAdvocate database from a data model perspective to establish a preliminary design based on relationships and needs of the users. Then, we will gather functional dependencies about the data and use the dependencies to decompose the database into an optimized normal database.

Looking at semantic data models and entity relationship models allows for an early conceptualization of the database based on information gathered from users and developers. This conceptualization will allow developers to get a better understanding of data that is stored in the database and the possible relations that can exist (Kay, 2003).

To a certain extent, the major of the work in developing semantic data models has already been stated in the previous section. The storage needs for the users' background and other related information plus tracking information for pages viewed are very well specified. Just to reiterate the relations between the data variables, the following are some ER models for the MyAutoAdvocate site.




The is a ER model for data to track the pages that a user views




This is an ER model for the data to track the reward points and redemption

With developed ER models for the data, the next step would be to the convert the ER models to a relational data model. The conversion process and description of the relational model are described in more detail in the Intel project section; but, in general, without any constraints the mapping to a relational data model is very straightforward. Every attribute in the ER model would simply map to an attribute in the table of the relational model (Halpin, 2001). The following SQL commands creates tables that capture the preceding information for many aspects of the MyAutoAdvocate data.

```
CREATE TABLE "User_Info" (
Harris_ID bigint(10)
Fname varchar(20)
Lname varchar(20)
Password varchar(20)
Email varchar(50)
Street varchar(100)
City varchar(50)
State varchar(50)
Zip varchar(10)
Gender char(2)
Age int(3)
```

```
Pref_Attrib char(2)
OS varchar(10)
Browser varchar(20)
ModemSpeed varchar(10)
Login varchar(20)
V_Brochure tinyint(1)
Date datetime
Pur_Period varchar(10)
Pur_Type varchar(10)
Survey_Source char(3)
Deactivated tinyint(1)
);

CREATE TABLE "Sites_Visited"(
Harris_ID bigint(20)
Site varchar(50)
Points int(3)
Date datetime
Activity varchar(50)
);

CREATE TABLE "Rewards_Redemption"(
Harris_ID bigint(20)
Pts_Earned int(3)
Earned_Amazon int(3)
Earned_Lottery int(3)
Proc_Amazon int(3)
Proc_Lottery int(3)
Fname varchar(100)
Lname varchar(100)
Email varchar(100)
);
```

## Functional Dependencies and Normal Form

Now that the MyAutoAdvocate data is in relation models, the next step would to be to reduce redundancies through decompositions using functional dependencies. Again, for a more in depth description of redundancy, why redundancy is so harmful for databases, and how to eliminate redundancy look at the previous chapter on the Intel project. The previous chapter will also include the exact algorithm for decomposing a database into Boyce-Codd Normal Form, BCNF, using Functional Dependencies, FDs. Therefore, this section will mostly describe the exact decomposition and the FDs that are

involved in refining the tables the MyAutoAdvocate data, and finally illustrate the database that has been optimized.

It would be really easy to make a big database that contains all the information for all users. Since the data for MyAutoAdvocate is centered on pages, it would be very simple to create a large table that records each page that user views. Each entry then would have to not only contain information on page clicks, but also information on the users and rewards. Although it would be very simple to setup, the drawbacks of this large table design due to redundancy are just too great (Roman, 1999). Therefore, the efficacy of a well organized database is very important. This is where decomposition and putting tables into normal form comes in. To start the decomposition process, a list of the FDs must be made. The FDs just list simple relationships between the data variable. For example, in the MyAutoAdvocate data, a HarrisID value can be used to associate a user with the user's background information. These associations help to ease the redundancy and help create tables that are in normal form (Janert, 2003).

The methodology of procuring FDs is not very difficult. Since FDs represent the relationships and constraints that exist between variables, it is usually very straightforward to determine FDs from a good understanding of the data (Zaiane, 1998). Here is a list of a few FDs for the MyAutoAdvocate data.

- HarrisID → Fname, Lname, Password, Email, Street, City, State, Zip, ,Gender, Age, OS, Browser, Modemspeed, Survey_Source, Date, Pur_Period, Pur_Type

- HarrisID → Pts_Earned, Earned_Amazon, Earned_Lottery, Proc_Amazon, Proc_Lottery, Fname, Lname, Email
- HarrisID, Date → Site, Points, Activity
- HarrisID, Date → Rearch_Site, SessionID
- HarrisID, Date → Brochure

With a list of complete FDs, it is now possible to decompose the data into a normal form, such as BCNF. The entire algorithm for decomposition is shown in the Intel chapter of this paper; please refer to that for a more a detailed explanation of the algorithm and normalization. Using the FDs that have been developed for the MyAutoAdvocate data, the following are a few tables have been reduced to BCNF form to store the data needed for the study. Please note that not all of the tables in the entire MyAutoAdvocate database are listed.

"Brochure Info"
Field Type Attributes
Harris_ID bigint(10) No
Time datetime No
Brochure varchar(200) No

"Pages Visited"
Field Type Attributes
Harris_ID bigint(20) No
Site varchar(50) No
Points int(3) No
Date datetime No
Activity varchar(50) Yes

"Login Info"
Field Type Attributes
Harris_ID bigint(10) No
Time datetime No
SessionID varchar(10) No
Login varchar(20) No
Connection varchar(10) No

"Rewards Redomption"
Field Type Attributes

Harris_ID bigint(20) No
Pts_Earned int(3) No
Earned_Amazon int(3) No
Earned_Lottery int(3) No
Proc_Amazon int(3) No
Proc_Lottery int(3) No
Fname varchar(100) No
Lname varchar(100) No
Email varchar(100) No

## File Organizations and Indexing

So far, we have only discussed redundancy and normal forms as the main ways of optimization, which mainly focus on storage efficiency and data correctness. Another aspect of data management that is interesting to investigate is speed of data retrieval. The speed that data is retrieved in a database figures directly into performance of all database actions. All operations ranging from simple searches to complex joins need to find specific entries in a database, and this becomes a more difficult task as the size of the database and the number of entries that needs to be searched increases (Owen, 1998). The following section will look at how this issue affects the MyAutoAdvocate database and what kinds of solutions are presented to optimize performance.

With about five thousand users, the size of the database that contains all the information for MyAutoAdvocate is considerably large. Without a systematic search process for a large database, straightforward tasks such as increasing the reward points of a user might prove to be costly. Giving more reward points to a user could require the system to search through the entire database of five thousand users to find the correct user. Doing this each time a change is made in reward points can incur a big running cost. An even bigger problem could arise from the data of pages that users have viewed.

Assuming that each user sees about ten pages, the table that contains the pages information could easily grow up to fifty thousand entries. Searching for all the pages that a specific user views could easily translate into searching through all fifty thousand data entries. This is huge cost considering this process needs to be done possibly five thousand times for each user.

To solve this problem, the use of indices is needed for the database. An index is a data structure that organizes data records on disk to optimize certain kinds of retrieval operations (Ramakrishnan and Gehrke, 276). Using an index, we can efficiently search and retrieve all the records that satisfy a certain search condition. In the MyAutoAdvocate data, HarrisIDs are a natural fit for an index. When information is stored in the MyAutoAdvocate database, a record is kept to track the memory that is allocated to each entry. The indices, which in this case will be HarrisIDs, are then placed into a tree structure, usually a binary tree or a K-Tree. This makes searching much easier. Advanced search algorithms exist for a tree data structure and these algorithms can drastically cut down search times (Garrick, 1997). The implementation on the data for MyAutoAdvocate allows for search times to operate in *log* time and provides for much faster data retrieval. This is a great optimization for the database and creates a much better performing database (Garrick, 1997).

## Conclusion

The Intel project and the MyAutoAdvocate project are very interesting case studies of data management in a real setting. The projects are a great opportunity to see the problems of data management. Clickstream data of the Intel project was a good example of redundancy of data. The Intel project illustrated how mismanagement data can cost dearly in physical space and this problem hurt the Hierarchal Bayes part of the analysis. The MyAutoAdvcoate project also demonstrated how the redundant data and costly data retrieval costs of the project. These two cases also shine light on how to apply some of the latest database optimization techniques. In MyAutoAdvocate project, we were able to apply a range of procedures that allowed us to reduce the amount of storage space and speed up the time needed to retrieve data. Working with the two projects has showed the importance of a strong understanding of the data and how this understanding from the start will solve many of the problems that can stem later. It was very interesting to apply theoretical concepts regarding database management systems in a real life scenario, and it was a pleasure to see concepts I have learned from class utilized. Working on the Intel project and the MyAutoAdvocate project has taught me a great deal on database management systems and it was great to see computer science benefiting other fields of research.

References

Tsou, Don-Min, Fischer, Patrick C., (1982) "Decomposition of a relation scheme into Boyce-Codd Normal Form", *ACM SIGACT News*, (Summer) 23-29.

Anonymous, (2002) "Internet Growth Now and the Future, 2002" http://www.dmnews.com%2Fcgi-bin%2Fartprevbot.cgi%3Farticle_id%3D24592

Mena, Jesus (2001), "WebMining for Profit: E-Business Optimization", Butterworth-Heinemann.

Bucklin, Randolph E., James M. Lattin, Asim Ansari, Sunil Gupta, David Bell, Eloise Coupey, John D.C. Little, Carl Mela, Alan Montgomery, and Joel Steckel (2002), "Choice and the Internet: From Clickstream to Research Stream", *Marketing Letters*, 13 (3), 245-258.

Silverston, Len (2001) The Data Model Resource Book: A Library of Universal Data Models for All Enterprises, Revised Edition, Volume 1, Wiley Publishers

Kay, Russell (April, 2003), "QuickStudy: DataModels", *ComputerWorld*

Vossen, Gottfried (1991) Data models, database languages and database management systems, Addison-Wesley Longman Publishing Co., Inc.

Hay, David (1995) Data Model Patterns: Conventions of Thought, Dorset House

Halpin, Terry (2001), Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design, Morgan Kaufmann

Naudé, Frank (2001) "General Database Design FAQ" http://www.orafaq.com/faqdesgn.htm

Janert, Philipp (2003) "Practical Database design, Part 2" http://www-106.ibm.com/developerworks/web/library/wa-dbdsgn2.html

Roman, Steve (1999) "Access Database Design & Programming, 2nd Edition" http://www.oreilly.com/catalog/accessdata2/chapter/ch04.html

Anonymous (2004) "Advanced Normalization" http://www.utexas.edu/its/windows/database/datamodeling/rm/rm8.html

Meloni, Julie (2002) "MySQL: Learning the Database Design Process" http://www.samspublishing.com/articles/article.asp?p=27281&seqNum=4&rl=1

Zaiane, Osmar (1998) "Functional Dependencies: Basic Concepts" http://www.cs.sfu.ca/CC/354/zaiane/material/notes/Chapter6/node11.html#SECTION00651000000000000000

Aaby, Anthony (2003) "Functional Dependency"
http://cs.wwc.edu/~aabyan/415/FunDep.html

Garrick, Joe (1997) "Jose's Database Programming Corner"
http://b62.tripod.com/doc/dbbase.htm#index

Owen, Cathy (Mar. 1998) "Data Modeling & Relational Database Design" *The IDUG Solutions Journal,* Volumne 5, Number 1