WEB ENGINE FOR INVESTIGATING CONSUMER CONSIDERATION

by

Eleojo E. Ocholi


Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 2009

© 2009 Massachusetts Institute of Technology
All rights reserved.

Author

Department of Electrical Engineering and Computer Science

Certified by

Professor Glen Urban
David Austin Professor of Marketing
Chairman, Center for Digital Business at MIT
Thesis Supervisor

Accepted by

Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses
WEB ENGINE FOR INVESTIGATING CONSUMER CONSIDERATION

by
Eleojo E. Ocholi

Submitted to the
Department of Electrical Engineering and Computer Science

May 22, 2009

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Today consumers are presented with a plethora of products each time they want to make a purchase. Sometimes they have up to thousands of options and configurations to pick from and yet many consumers are shown to initially screen this size to create a more manageable set to truly consider in an in-depth way. Companies today are looking for ways to ensure that their products make it into the smaller consideration sets of consumers in order to increase the probability of sales. This thesis documents the design of a web engine that provides a survey framework for investigating algorithms that aim to predict which products a user will place in their consideration set as well as to aid in investigating the factors that can lead to the modification of rules that govern a consumer's consideration set. Firstly I evaluated and documented the improvements required from older systems created by the research group. Then over the course of two studies I designed a highly modular system that is a new iteration of the older versions. Finally, more than 3500 participants used the system during field tests and the system was successful in mitigating the previous issues and delivering a better user experience as well as collecting the necessary data. This project lays the groundwork for a platform that can be used for generally investigation and testing consideration predictive algorithms in various retail spaces.

Thesis Supervisor: Glen L. Urban
Title: David Austin Professor of Marketing
ACKNOWLEDGEMENTS

Firstly, I would like to thank my advisor Professor Glen Urban for overseeing all the work I did. He has been a great advisor and mentor, providing me with a great opportunity and invaluable skills that will continue to benefit me in my career and life. I would also like to thank General Motors for making the project financially possible as well as being a pleasure to work with.

Special thanks goes to Rene Befurt and Erin MacDonald for managing and supporting me during the project. I learnt a lot from you. Also many thanks to Kevin Wang, Daria Silinskaia and Jong-Moon Kim for being so fun work with and for contributing greatly to the success of this project.

Finally, I thank my parents and brothers for the love, commitment and support over my entire life and especially the last five years at the institute and I thank my Lord and savior Jesus Christ for giving me a reason to live. To all the people I missed out from this list you know who you are and I am greatly indebted to you... Thank you!
# CONTENTS

Acknowledgements .................................................................................................................................................. 4

Chapter 1 Introduction ........................................................................................................................................ 8

1.1 Summary of Thesis Content ....................................................................................................................... 11

Chapter 2 Background ...................................................................................................................................... 12

2.1 Previous Systems ........................................................................................................................................ 13

2.1.1 Bullpen/Countdown Task version 1 ........................................................................................................ 13

2.1.2 Card-sort Task ....................................................................................................................................... 15

2.1.3 Bullpen Task version 2: (First new iteration) ......................................................................................... 17

2.2 Content Generation/Management: Model-View-Controller Paradigm .............................................. 19

2.3 AI Predictive Algorithm ............................................................................................................................. 19

Chapter 3 Overview of System Design ........................................................................................................ 21

3.2 Summary of Changes ................................................................................................................................. 22

Chapter 4 Architecture and Implementation ................................................................................................ 24

4.1 Main Survey Framework .......................................................................................................................... 24

4.1.1 Testing Inductions ................................................................................................................................. 27

4.2 Main Consideration Task: Countdown .................................................................................................... 28
4.2.1 Flexible design of profiles ................................................................. 31
4.2.2 From Django to Flash: XML ............................................................... 32

Chapter 5 Evaluation and Discussion .......................................................... 34
5.1 Results and Contributions ........................................................................... 34
5.2 Future Work ............................................................................................... 36

References ....................................................................................................... 37
List of Figures

Figure 1: Screen shot of first Bullpen implementation ........................................................ 14
Figure 2: Screen shot of new Flash-Based Bullpen ..................................................................... 18
Figure 3: Overall System Diagram .................................................................................................. 25
Figure 4: Control Flow Diagram of user click in Survey Framework ........................................... 26
Figure 5: Consideration Task Control Flow Diagram ...................................................................... 28
Figure 6: Implementation Structure of Main Consideration Countdown Task................................ 29
Figure 7: Sample Product Profile .................................................................................................... 30
INTRODUCTION

In the field of marketing, researchers focus on developing new strategies to increase sales for businesses. An important aspect to this is trying to understand how consumers make decisions about which products to buy. That is, what process do they go through when they consider products? What features of products matter and how can we predict what a user will choose to consider? This project aims to aid in answering these questions. The more the process of consideration is understood, the greater the ability of the business to influence a consumer’s decision process, thereby increasing sales.

When consumers need to choose between large numbers of products, particularly in today’s online-retail environments, they usually screen the alternatives and create a smaller set of products, which they will seriously investigate and proceed to choose from (e.g., Hauser & Wernerfelt 1990). This is also known as a two-stage consider-then-choose decision process. The smaller consideration set varies between product spaces. For example, consideration sets for packaged goods are typically 3-4 products rather than the 30-40 products on the market (Hauser & Wernerfelt 1990; Urban & Hauser 2004) while in the automobile space people decide which 10 or less vehicles to choose from out of more than 350 make-model combinations of products.
The quality of the products contribute to their selection in these sets but in order to be selected, it is not enough to create a good product that is top quality. For example, General Motors has invested heavily in product design and quality such that in 2007 Buick tied Lexus for the top spot in J. D. Power’s vehicle dependability ranking, in 2008 Buick tied Jaguar for the same top spot and also in 2008 Buick was the top US brand in Consumer Reports. Despite this, “... roughly half of US consumers (and 64% in California) will not even consider a Buick” (Hauser, Toubia, Evgeniou, Befurt, & Silinskaia, 2009). General Motors would like to understand what affects a consumer’s consideration set and discover what could be done to modify such sets.

In the area of decision-making a lot of research has gone into studying the rules that people use in order to decide which product profile to choose from multiple attribute/feature configurations. There are several rules that subjects could use in making the decision and the assumption in random utility models used to model such rules was a compensatory model, where it is assumed that subjects are able and willing to make trade-offs between attributes in order to determine the most preferred alternative. That is, subjects use compensatory heuristics whereby attributes of a profile are weighted by their contribution to the subject’s utility in order to evaluate the relative utility of that profile, and eventually they choose the profile with the largest utility value. Yet research has shown that people don’t necessarily like making trade-offs and non-compensatory rules also fit the decision making process (Payne, Johnson, Bettman, & Coupey, 1990) particularly in complex products categories. The two-stage, consider-then-choose decision process explained in the previous paragraph is a particularly relevant model when considering a large number of product alternatives. It is reasonable to assume a non-compensatory...
consideration stage in such a situation and predicting such non-compensatory decision rules in the consider stage of this two-stage process has been a recent focus of research at the Center for Digital Business. (Hauser et al, 2009) and the different systems built to investigate consideration.

In order to investigate consideration and its influencing factors, I re-designed and developed a web based measurement tool/engine. In a multistage approach, the engine uses a user adaptive machine-learning algorithm to find revealed preferences and build up conjoint-like tasks, stage by stage based on these estimates. This procedure is a challenge since many operations are supposed to happen on the fly. Such a task calls for very efficient programming and data handling as well as providing a similar experience for the various browsers out there in the online user space.

This past year, GM signed on with us to conduct significant studies into customer consideration. The measurement tool in the form of a web survey-framework and corresponding modules was used to examine and explore the way users consider cars and possible ways to influence auto-consumers in order to improve the consideration of car models from this manufacturer. The purpose of this Thesis was to improve the design and implementation of the existing consideration ‘Bullpen’ task and to create a system using a newly developed adaptive algorithm for predicting consideration rules and sets. This document will describe the issues with previous system designs, mitigation of those issues,
implementation of a new system as well as the evaluation of running the new system with thousands of real-world users.

1.1 SUMMARY OF THESIS CONTENT

Chapter 2 will provide the background information about previous iterations of systems built to solve the problem as well as their limitations.

Chapter 3 will give an overview of the final system designs for the engine and enumerate design decisions.

Chapter 4 will go into details about the architecture and implementation of the web engine.

Chapter 5 will discuss the technical evaluation of the system after running studies using them with real-world users as well as personal contributions and future work.
CHAPTER 2

BACKGROUND

Consumer consideration is important. When people shop, they tend to analyze only a small subset of many models on the market. If a particular product is not in the consumer’s “Consideration Set”, it will be eliminated from contention. The user is not likely to notice advertisements or information for non-considered products. In order to investigate how these sets are formed and affected, we need to focus on one area but must build systems that can be generalized.

The online retail space is quite vast so the system described by this thesis was created using the automobile space, but can be modified for use in a more general context. In the car market, the consideration of vehicles from US car manufacturers is low (Less than 50% -- 36% in LA). Thus they would like to understand more how the consumer makes choices about which cars to buy and hopefully influence the decision process to include more of their cars.

The following sections describe some of the previous systems built by the Center for Digital business to collect data used later to model decision rules (Hauser et al. 2009) for forming consideration sets.
2.1 Previous Systems
The need was to create an online measurement tool to collect data and to be able to estimate the elimination/inclusion rules used by a consumer when choosing a vehicle. Thus research assistants in the Center for Digital Business, created online systems based on a methodology decided by researchers to investigate the rules used by consumers when choosing products. That is, systems used to collect consideration data for use offline in estimating rules. The main consideration tasks were placed within surveys and two field tests were run: one with GPS devices as products and one with cars. The systems worked quite well but unfortunately, the efficiency, flexibility and compatibility of the design of the systems were severely constrained. The compatibility issue was fixed first and a system using the established survey design was created, but the design of the module that contained the main consideration task was changed. Finally, another field pre-test was run with the change. Discussed below are the 3 previous systems created for the consideration tasks used in field tests. Previous Research Assistants, Clarence Lee and Shirley Fung implemented the first 2 systems. I implemented the third system discussed in this section.

2.1.1 Bullpen/Countdown Task Version 1
A major part of the consumer experiments involve “bullpen” and “countdown” tasks, where users have to choose which product models they would consider out of a large collection of models. Figure 1 below shows a screen shot of a bullpen from an earlier study customized for choosing GPS devices.
The technical design of the tasks used frames for the major panes and used JavaScript calls to refresh all the panes individually when an event is called.
Shortcomings:

The system needed to work on all browsers including older versions. Hence the JavaScript calls used needed to be old ones that all versions of browsers would understand and implement the same way. Being limited to these calls caused the implementation to be rather "clunky" and not very efficient.

Refreshing each frame per click also wasted bandwidth. Unfortunately due to the JavaScript and browser compatibility mentioned above, this was the reliable way to ensure all users had the same experience.

Finally the different functions and parts of the system were highly coupled together making it difficult to upgrade different parts of the system or easily change a small part of it. There was no modularity

2.1.2 Card-sort Task

This task required users to sort a deck of cards online as they would in real life. The cards showed car images and attributes of the various makes and models. The user was required to place cards in piles; a pile for those that the user would consider and a pile for those that

---

1 JavaScript methods do not work the same way on all browsers. Especially old ones, and it is difficult to predict the behavior of the system for all browsers. Not all users have up-to-date browsers that implement JavaScript calls the same way so when using this language there is an inherent risk
the user would not consider. In order to improve the user experience, the task used many
2 AJAX calls.

**Shortcomings:**

The card-sort task shared the same problems as the bullpen task system above. In addition,
there were significant issues based on the different implementations of JavaScript on
different browsers. A lot of user data was lost or corrupted due to unanticipated behavior
of the system on various browsers. Eventually the study excluded many users in order to
support a small subset of browsers.

Also the task required users to drag cards across the screen to different piles in order to
mimic the task in real-life but the user comments on the system showed that they greatly
disliked this.

2 AJAX - generally stands for Asynchronous JavaScript And XML. Used in most web 2.0 sites when creating
user interfaces and thus improve the overall user experience by providing lots of functionality for user
interaction.
2.1.3 Bullpen Task Version 2: (First New Iteration)

There were quite a few issues from old systems that needed to be fixed in the new iterations, so the changes were tackled over 2 field studies. The first change tackled was the compatibility issue as the lack of JavaScript compatibility caused the most damage to data collected. The first design decision was to implement the main bullpen task using ActionScript 3.0 (AS3), an Adobe Flash technology. AS3 requires the Flash Player 9 virtual machine to work, so it behaves exactly the same way on all browsers because the same company implements it for every browser. Also compared to other online technologies, the performance was shown to be better than JavaScript and previous versions of ActionScript (oddhammer.com viewed June 2008). Finally the reported penetration numbers for the required Flash Player 9 in North America, at that time according to Adobe, was greater than 95%, which was enough for our field tests. The Flex programming API was used to implement the new bullpen in AS3.

Advantages of new implementation:

- Faster
- Communicated with the server fewer times, thus reducing bandwidth used.
- Smoother user experience
- Improved Aesthetic appeal compared to previous versions.
**Shortcomings:**

The entire survey was still highly constrained and not flexible or easily modifiable. Also it could only communicate with server at the beginning or at the end of the task, thus information loaded into the client browser at the start could not be changed by the server until the end of the task. This problem would need to be solved to provide flexibility for predictive technologies coming in the final version.
2.2 CONTENT GENERATION/MANAGEMENT: MODEL-VIEW-CONTROLLER PARADIGM

In the previous section, I discussed the previous systems that researchers, in the Center for Digital Business, built for studying consideration. Now I will discuss the main design paradigm used for the new implementation. For interactive applications, A Model-View-Controller design pattern works to divide up the workload on each component and serves to decouple the functions of a system. The Model is responsible for storing the data and providing methods to the controller for accessing it. The View is responsible for presenting the data to the user on screen and the Controller handles receiving the data and communication passing the data between the view and the model. This allows the user interface front-end to be separated from the application backend encapsulated in the model. Thus they can be changed independently and the same model can be used with many views into it. Using this design pattern was the key to creating a flexible and customizable design that can easily be extended in the future. In their paper on web-development using the MVC pattern, Leff and Rayfield (2001) describe the issues involved in applying this pattern to Web applications- particularly the inherent difficulty in truly partitioning the model and controller portions of the system due to the client-server partition in web systems.

2.3 AI PREDICTIVE ALGORITHM

The introduction of this document briefly summarizes the theory and importance of being able to predict consideration sets in retail spaces. As the number of features for a particular product-type increases, the number of profiles to choose from scales up as well. Simply asking a user to evaluate every possible profile may require them to view tens of thousands
of profiles in one sitting, which is effectively infeasible. For example in the US automobile space, the many features to consider would have a respondent trying to screen more than 13,000 profiles! Hauser and Silinskaia developed an adaptive algorithm that creates a decision rule for each respondent using a ‘seed’ profile that the user configures. Once the profile is configured, the algorithm generates a small number (10 – 30) of questions or profiles that enable it fine-tune the decision rule for that user, allowing it predict which profiles in the entire profile space the user will consider. “After the user classifies a profile as considered or not, the adaptive question design algorithm (1) performs a Bayesian update of the decision rule, with priors based on the configured profile and market level data, and (2) generates a profile to obtain maximal information. By optimizing each data point, we are able to learn the decision rule in far fewer questions than an orthogonal design would require, making the task feasible for a respondent.” (D. Silinskaia, Personal Communication, May 17, 2009). The actual implementation of the algorithm was done in C++ because of the complexity and thus the amount of time it takes for it to generate the next question to be asked. The C++ code could be compiled into a dll that in general is executed much faster by the processor and can be called by multiple other technologies. Another Research Assistant, Paul Tsier implemented the C++ module.
CHAPTER 3

OVERVIEW OF SYSTEM DESIGN

The design strategy was to build upon a framework, and create a web system & modules for running studies to investigate consumer consideration. The chosen framework needed to satisfy certain requirements: (1) It needed to be MVC based, (2) open source, (3) have a strong following in web software development circles with adequate documentation, (4) be implemented in a language which has lots of libraries that can bridge easily with ActionScript and with C++, (5) be a computationally efficient & fast language and (6) provide a good test framework as well.

Based on the above requirements the list was narrowed down to PHP-based and Python-based frameworks. After further investigation a Python-C++ bridge was found that really helped with fulfilling the 4th requirement above. Thus after considering the various web-frameworks, the Django framework based on the Python language was selected. Django framework is an open source framework that follows the MVC model, and allows users to quickly prototype Web applications in python and any database of choice. Also a Postgre SQL database was used because it has the most support on the Django platform. An additional benefit of the technologies selected is that there are extensive modules, utilities and libraries provided in python and Django that help with many programming tasks.
3.2 Summary of Changes

With the research moving on to the next phase, the surveys and tasks were upgraded in these major areas.

- The generation of new profiles by the algorithm needs to be a real-time process, and the consideration task needs to be easy to use.

- The Surveys were changed to allow the introduction and testing of factors that influence the rules used by a consumer in forming a consideration set as well as the factors that can lead to changing the set.

- Compatibility: Browser compatibility had to be assured based on problems in the card-sort run. The pre-test run of the new bullpen confirmed the mitigation of this problem, by using a Flash/ActionScript based task.

- Flexibility: The initial implementation of the survey tools/tasks were upgraded using the MVC design pattern to improve performance, flexibility and create customization capabilities across product retail spaces.

- Extensibility: The system was upgraded to be able to use an algorithm that creates rules and predicts, in real-time, which products the consumer will consider buying. It is set up to be able to switch algorithms without having to make too many modifications to the system. All that is required is to write a new python module with the correct methods.

- Improved-Communication: The old flash module used xml to communicate with the server only at the beginning and end of the task thus improving bandwidth usage but
flexibility of design was still an issue. The latest module has improved flexibility allowing communication during the task but still reducing bandwidth by using a binary protocol for transferring the xml data.
CHAPTER 4

ARCHITECTURE AND IMPLEMENTATION

This chapter describes an overview of the entire system implementation. The basic experimental setup for a field study remained unchanged from previous studies. The entire setup is an online survey with a major consideration task within it. The user usually goes through preparatory pages before performing the task the first time. The system then shows mind-cleansing activities to the user before making them go through the consideration task once more. Those activities may also be used to test how a piece of information could affect the users consideration set in the second attempt.

In this implementation, the system can be broadly divided into the survey framework and the consideration task module.

4.1 MAIN SURVEY FRAMEWORK

The Survey framework is effectively the Django framework with methods added to the main module that acts as a controller in order to improve flexibility and customization power. The general Design of the system is shown below in Figure 3, particularly arrows 1 – 4.
Figure 3: Overall System Diagram.

Arrows 5 & 6 are only part of the Main Consideration Task Module
Figure 4 above shows the flow through the system after a user click. When the user clicks the 'Next' button on a survey page an HttpRequest is created and passed to the 'urls.py' module in the Django framework, which routes requests to the specific methods inside the module 'views.py'. This python module communicates with the 'models.py' module in order to perform calls to the database. After this, the method executing in views.py returns with a call to a template. Finally the response constructed using that template is sent to the user's browser.
M-V-C Structure and Flexibility:

Django is created with a tweaked version of the MVC design pattern (Gamma et al, 1995) but the naming conventions in Django are quite confusing. This is because the controller is actually called 'views.py'. While the equivalent of a view in the MVC design pattern is called a template. The file called 'models.py' controls the models and database access.

An addition to the flexibility provided by MVC, the structure of the views.py file makes it easier for a developer to add and remove pages to the survey. Every URL request to the survey calls the same method called 'survey(request, page_id)'. This method takes in the page_id as a parameter and acts as a switchboard dispatching different commands for different page_id's. The design assumes the pages follow in order with each page-id a number, thus the first page corresponds to page id 100. And the second page is 200 and so on. The templates are called survey100.html, survey200.html and so on.

4.1.1 Testing Inductions

A major requirement of the new system was to be able to allow the introduction and testing of factors that influence the rules used by a consumer in forming a consideration set as well as the factors that can lead to changing the consumer set. These inductions take the form of online blogs, articles, mind-cleansers and so on. In general a consumer reads the information presented to them and then enters a response(s) to the information. Due to the flexible nature of the survey design and the homogeneous nature of the collected data (large pieces of text), the system easily accommodates. When every respondent starts the survey they are assigned to a cell of users. Based on this cell, the same template will display different articles or blogs and so on. Just one template file needs to be changed to modify
the various inductions being tested. Currently the survey system is set up to test up to 2 inductions.

4.2 MAIN CONSIDERATION TASK: COUNTDOWN

The main focus of every survey is the main consideration task. In previous surveys, a bullpen was used but in the current system, a countdown exercise was chosen.

![Diagram showing the control flow of the main task] Figure 5: Consideration Task Control Flow Diagram

Figure 5 above shows the control flow of the main task. This flow is implemented as a module, created with Flex 3 & ActionScript3. This module communicates with the Web engine code in the Django framework via an open source set of python modules called
PyAMF\(^3\), which translates between python and AMF (a data format that other programming languages can use to communicate with ActionScript). PyAMF contains a special Django module as well. Below is a diagram that shows the structure of the Main consideration task implementation.

![Diagram of Main Consideration Countdown Task Implementation](image)

**Figure 6:** Implementation Structure of Main Consideration Countdown Task. Solver.pyd and the Boost_python dll perform the adaptive algorithm calculations.

When the main consideration task template is returned to the client browser and evaluated, a call is made to the server to download the flash module. On loading, the flash module makes an RPC (remote procedure call) to the server and requests a DjangoGateway.

---

\(^3\) More information on PyAMF can be found at [http://pyamf.org/]
object (created by in the PyAMF module). When the browser receives the object, the module and the server can communicate back and forth via binary, which greatly improves the performance over text-only-based protocols like XML or SOAP. The python module called ‘Gateway.py’ contains the implementation of the methods for communication with the flash module.

For an arbitrary number of questions, a profile is presented to the user by the module and the user considers or rejects the profile shown. A profile is a just a collection of attributes which define a particular product in the retail space. An example of a car profile presented to a user is below.

![Sample Product Profile](image)

**Figure 7: Sample Product Profile**
Each time a decision is made, the answer is sent back to the server and the application decides what next profile to show the user. If we want to use the algorithm, the answer is passed to `solver.pyd`: an external python module and it calls a dll that runs the algorithm and returns the next question. The algorithm was developed by Daria Silinskaia and for performance reasons, was implemented in C++ by Paul Tsier. The implementation includes settings (contained in 'Gateway.py') that allow the developers decide which questions to get from the algorithm module and which to get from another predefined source.

At the end of the countdowns, the Flash module presents the users with all the profile cards they chose and asks them to rank them. Finally 100 points are divided between the top 4 profiles ranked and then all the information is saved to the database by the gateway and the next page of the survey is requested.

### 4.2.1 Flexible Design of Profiles

Although we used automobiles for our field test and research into consideration, the system can be used generally in the consumer space. Firstly, the system assumes every profile has an image associated with it. Each profile also has/is a collection of attributes (also called features) and each attribute was a number of feature levels. For example, the ‘Engine Type’ attribute has 2 feature levels: ‘Gasoline’ and ‘Hybrid’. The different attributes and features are defined in a python module called features.py. The method `current_features()` returns a python dictionary where each line has the general form `key = integer & value = tuple of properties:`
Attribute_number: (‘attribute_name’, ‘display_type’, [‘level1’, ‘level2’, ‘level3’,…]).

For example,

```plaintext
  4: (‘Engine Type’, '', [‘Hybrid’, ‘Gasoline’]),
```

In the example above, Engine Type is the 4th attribute and with the display_type empty, it defaults to a String. Finally the different levels are ‘Hybrid’ and ‘Gasoline’. For the image associated with a profile, the key in the dictionary is the String ‘img’ not the Attribute_number. And the value is a list of filenames. The developer can decide how to use the order of the filenames. For example in the automobile case we have:

```plaintext
  'img': ['sportscar.png', 'hatchback.png',...]
```

The order of filenames is in the same order as the first attribute called ‘Body type’. There is code in the gateway module that uses that order similarity to match images with the relevant ‘Body type’. Moving forward, the method that creates profiles can be tweaked to define how to assign images to profiles.

### 4.2.2 FROM DJANGO TO FLASH: XML

When the gateway is first loaded, the features are loaded in as well. So when the flash module requests a profile, the gateway converts a list of zeros and ones into a profile in XML. This xml string is converted to binary and passed to the flash module via the connection established by PyAMF. The structure of the xml is shown below:
Multiple <feature> tags are placed within <FeatureMap> and the label is what is displayed on the profile card in as the label for an attribute. The display tells the flash module how it should display. For example if display is set to ‘stars’ the flash module expects the level value to be an integer from 0 to 5 stars and the corresponding number of stars is displayed. As the project moves forward more ways of displaying information can be added to the flash code to make it more flexible. Also, even if the gateway python code is re-written, as long as the xml structure is preserved, the module can be easily reused.
CHAPTER 5

EVALUATION AND DISCUSSION

5.1 RESULTS AND CONTRIBUTIONS

Previous research has shown the theory and advantages to being able to predict consideration sets for consumers in a retail space. In this thesis I have described an implementation of a system that can be used to run studies as surveys using the algorithms developed to predict such consideration sets.

The current implementation uses an algorithm developed by Hauser and Silinskaia to discover and predict the consideration rules for respondents, particularly in the automobile space. The survey system was tested on 4000 real world users split between 3 servers. The flexible design of the system along with good development practices allowed us to begin development even before the final survey design was finished in order to meet a deadline set by the research sponsors. Certain design choices did not make it into this implementation and I will discuss some those choices in the next section on future work.

That being said the current system incorporates all the changes required as described in section 3.2. As mentioned above, after the implementation of the Web engine, the system was tested with real users and it was able to handle a large volume of real-world users and run a complicated algorithm while delivering a smooth user experience. Over this experience a few lessons have been learned. Firstly it paid off that a lot of time
was dedicated to the initial design of the final implementation because this allowed the survey content and flow to be designed in parallel with the implementation of the system in order to make a deadline. Another good decision was the use of version control software while implementing. This allows easy integration of multiple changes to different sections of the system made on different machines. More importantly, it allows the tracking of specific changes and aids in debugging problems.

During the course of the project, coding started even before the final design of the survey was finalized, thus changes were being made up until a few days before launch. One of those last minute changes I made wasn't complete and led to an error with the data from the validation group of the study. After discovering the error, we ran that portion of the study again with a thousand users and got good data. This experience underlines the necessity of freezing code changes multiple weeks before hand. If a change is made, then the study date should be pushed back or the entire testing process should be restarted from scratch for that feature. It also highlights the usefulness of having an automated test suite. This feature in question was something we had tested in previous weeks, hence after I made the changes, we didn’t do and in-depth test of the area as I had other changes to make to meet the deadline coming up in less than a week after that. Had we designed an automated test suite, we would probably have caught the error. Although creating an automated test suite takes a lot more resources and time, the potential benefits a definitely worth the investment. In the next section I highlight more areas to invest time in for future work.
5.2 Future Work

Although this current implementation achieved its primary purpose, changes will need to be made in order to transform it into a reusable platform for testing algorithms in consideration experiments. Such changes include:

- Creating a formal API to improve flexibility for substituting algorithms, to guide the creation of flash modules and so on.
- Creating a user interface for defining and loading into the system profile features and levels
- Adding more functionality to flash modules to increase forms of displaying profile data.
- Adding more consideration tasks (in addition to the countdown and bullpen tasks) for survey designers to choose from.
- Creating a User Interface for handling data and changing system settings for example, the number of questions to ask, which questions to get from the algorithm vs. some other defined source and so on.
- Defining guidelines for algorithm implementation in order to maximize the number of respondents who can perform the main consideration tasks simultaneously
REFERENCES


