

SUE:

**An Advertisement Recommendation Framework
utilizing Categorized Events and Stimuli**

by

Billy Chi Hong Cheung

B.Sc., The University of British Columbia, 2005

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Computer Science)

The University Of British Columbia

(Vancouver)

April 2008

© Billy Chi Hong Cheung 2008

Abstract

With the emergence of peer-to-peer video-on-demand systems, new avenues for keeping track of and subsequently meeting user needs and desires have arisen. Based on the idea of contextual priming, we introduce a new framework, SUE, that takes advantage of the intimate level of user profiling afforded by the internet as well as the linear and segmented nature of p2p technology to determine a user's exact on-screen experience at any given time interval. This allows us to more accurately determine the type of information a user is likely to be more receptive to. Our design differs from other existing systems in two ways: (a) the level of granularity it can support, accommodating factors from both the user's on-screen and physical environment in making its recommendations and (b) in addressing some of the shortcomings seen in current applications, such as those imposed by coarse user profiling and faulty associations. In order to examine the viability of our framework, we provide a high level design specifying its incorporation into an existing p2p video system, the BitVampire project.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
Acknowledgements	viii
Dedication	ix
1 Introduction	1
1.1 Motivation	1
1.2 Goal	2
1.3 Thesis Contributions	3
1.4 Thesis Organization	4
2 Background information and Related Works	5
2.1 Contextual Information	5
2.1.1 Background	5
2.1.2 Context, Feedback and Implication	6

iii

Table of Contents

2.1.3	User environment as a context	7
2.2	Related Works	7
2.2.1	Online advertisement	7
2.2.2	Psychologically Targeted Persuasive Advertising	8
2.2.3	Collaborative and Content-Based Recommendation Systems	9
3	System Design	11
3.1	Introduction	11
3.2	System Architecture	12
3.2.1	Advertisement Decision	12
3.2.2	Advertisement Storage and Retrieval	17
3.2.3	Advertisement Presentation	23
4	Implementation Analysis	25
4.1	Introduction	25
4.2	Clip and Video Metadata	27
4.3	Advertisement Components	29
4.3.1	Advertisement Decision	29
4.3.2	Advertisement Storage and Retrieval	33
4.3.3	Advertisement Presentation:	35
5	Evaluation	37
5.1	Simulation Setup	38
5.1.1	Simulation Parameters	39
5.2	Simulation Results	40

Table of Contents

6 Conclusion and Future Work	44
6.1 Conclusion	44
6.2 Future Work	45
Bibliography	49
 Appendices	
A Pseudo-Code	55

List of Tables

3.1	User Interest...	14
3.2	General User Interest...	15
3.3	Advertisement Objects...	18
3.4	Advertisement Storages...	19
3.5	Master Advertisement List...	20
4.1	Structure for metadata...	27
4.2	Structure for Events...	28
4.3	Structure for Movie modifier...	28
4.4	Structure for GeneralInterest...	30
4.5	Structure for Environmental Factors...	31
4.6	Structure for Advertisement Objects...	33
4.7	Structure for Advertisement Storages...	34
4.8	Structure for Master Advertisement List...	34

List of Figures

3.1	Flowchart for indicated interest	13
3.2	Types of networks	20
4.1	Overview of Advertisement Decision System	29
4.2	Sequence Diagram - Indicated Interest	32
5.1	Stimuli numbers vs. Average requests per minute	41
5.2	Event Frequency vs. Average requests per minute	42
5.3	Stimuli % vs. Average requests per minute	43

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dr. Son T. Vuong for his understanding, support, and the many opportunities he provided. I am thankful for the freedom he has given me in pursuing whatever topic I find interesting, offering suggestions and recommendations at every juncture while putting up with my many topic changes.

To my second reader, Dr. George Tsiknis, I offer my sincere thanks for the suggestions and insights he offered. Given his impossibly busy schedule, the time and effort he expended to help improve my thesis is doubly appreciated.

To Dr. Kurt Eiselt and Dr. Yvonne Coady, the perfect arguments against anyone claiming that TAing for the department is a pain. As disturbing as it may seem, I think I actually had fun.

To Ms. Hermie Lam and Ms. Joyce Poon, for all the help they have provided and for the forbearance they've shown in answering my many many many questions.

To the denizens of NIC lab, both the transients and the more or less permanent residents, here's a thank you that's been long overdue, not so much because I'm an ingrate but rather that there has never been a good opportunity for it. So, I hereby gratefully acknowledge all of you. Thanks.

Dedication

To the day that forced open my eyes. Should I thank you for teaching me why?

- Billy Chi Hong Cheung

April 2008

Chapter 1

Introduction

1.1 Motivation

Peer-to-peer(p2p) systems are traditionally considered as only a means of data sharing, often overlooked as also a medium in which end-users spend a lot of time with. Increasingly, these systems are moving away from being just a passive tool to facilitate file sharing[1][2][3](i.e. using the system to download a file found on a tracker, and then using another program, such as a media player, to access the file) to a more interactive agent, where the entire process of seeking, retrieving and viewing the file is available in the same application[4][5][6].

Such systems often maintain information about user activity[7][4] and can offer an intimate and previously unattainable level of customization through which a user's choices and preferences can be used to further supply them with advertisements targeted specifically for their apparent needs/taste. This becomes even more significant when one compares the granularity possible with this type of targeting to that available in other media of similar scale. (i.e magazines and web page advertisements are relevant only to their immediate content, TV ads likewise are based on target audiences of the genre of the show being broadcasted and/or the channel being broadcasted

on, and e-mail spam is done indiscriminately at best, and usually offers completely unnecessary products.) Yet we have yet to see anything that takes advantage of this new medium beyond targeting based on user selection of channels or movies[6], or through the use of collaborative filtering[7][8]. We aim to address this through the introduction of a framework that allows for a much finer level of granularity with which to track and make use of users' activities to provide them with recommendations.

1.2 Goal

The goal of this thesis is to examine the viability of a framework that explores a new paradigm to provide users with relevant recommendations. Instead of relying strictly on what the user thinks (such as use of user ratings) and what they state to be their preferences (such as the channels they subscribe to) we focus on what users do and what they feel. As such, we name our framework SUE (System Utilizing Empathy).

Our new framework allows user recommendations to be made based on three chief criteria: (a) what the user has been experiencing in the immediate past and the stimuli these past events generate, (b) the user's distant past history and inclinations, and (c) the physical environment the user is in. Given the sequential nature of video viewing (as opposed to, for example, text browsing, in which a user's focus can jump rapidly across any particular area of text), the exact on-screen content a user will be exposed to is much easier to gauge. As such, we focus the design of our framework towards online video (i.e. VoD) systems.

It is worth noting that this thesis offers just a framework for solving this problem and has no intention of providing a fully functional decision matrix that can mine and analyze user preference in a sophisticated manner, or deal with the issues concerning advertisement presentation in any great detail, both of which being topics for future investigation. In this work, we aim to prove the viability of such a framework in terms of its applicability into existing systems, and will include a basic advertisement decision system to determine what information is presented to the user through the use of weights from the aforementioned factors to decide advertisement category. To further explore its feasibility, we examine the implementation considerations surrounding the incorporation of our framework into a preexisting p2p video streaming application.

1.3 Thesis Contributions

The main contribution of this thesis is to allow for better targeted information to be provided to end users in a system that supports advertisement and maintains information about the content/service being provided. To achieve this, we:

- Offer a new framework to categorize the user's on-screen and environmental experience to determine the type of targeted information to send to the user. Our framework differs from other existing systems in the level of granularity it provides as well as the factors it takes into consideration in determining what to recommend to the user.
- Explore a new technique for determining the user's exact on-screen

experience at any given time during their use of the application, specifying the elements of a user's experience by dividing it into various stimuli, and ranking each event's stimuli.

- Introduce the concept of user environment as a factor in determining what type of data to push to the user
- Provide a high level design specifying the incorporation of the framework into an existing p2p video system, BitVampire.

1.4 Thesis Organization

This thesis is divided into six chapters. Chapter 2 contains background information and related works concerning the state of contextual information use with regards to their advantages and limitations, as well as how our approach differs from them. Chapter 3 translates our goal into a more tangible form, providing a proposed conceptual design, system architecture, and common structures. Chapter 4 applies our design into the existing BitVampire system, and shows how and where modifications should be made to incorporate our framework. In Chapter 5, we evaluate our framework in terms of how changes in the granularity with which it tracks user activity will affect the workload of an existing system. Chapter 6 will examine our results and what conclusions we can draw from them as well as discuss the many remaining issues for future consideration.

Chapter 2

Background information and Related Works

This chapter presents some basic background information concerning contextual information and their applications with regards to advertisements, as well as related works in the field.

2.1 Contextual Information

2.1.1 Background

The idea of making use of the context, or situation, in which one is placed as a means of garnering information about oneself is not a particularly new or innovative idea. In fact, it is readily applied in a variety of fields ranging from architecture and city planning to ubiquitous mobile devices detecting wireless peripherals and displaying the appropriate interfaces to normal everyday conversation.

In the context (pun not intended) of this chapter, we provide some background information on contextual information as it pertains to the impact of media context and advertisement type, as well as how the environmen-

tal context of a user can affect their mood, inclinations as well as shopping desire.

2.1.2 Context, Feedback and Implication

The priming principle[17] is a concept in advertising which states that different types of advertisements are perceived by consumers with varying intensity based on the context in which it is experienced. The principle asserts that the susceptibility of the consumer to a particular type of advertisement depends on the context that is serving as a primer. Context, as defined by [18], is the characteristics of the content of the medium in which the advertisement is inserted. The mood congruency-accessibility hypothesis[16][31] therefore suggests that advertisements that are relevant for or congruent with the mood of a subject at the moment may be more easily accessed and processed, with the implication that depending on the type of advertisement and the context, certain desires/needs can be made more salient.[18][26]

Whereas [39] has shown the viability of contextual priming on the textual aspects of advertisements, [34] examined the viability of the visual component, both reaching the conclusion that there is indeed evidence for priming effects on the visual aspect of advertisements. Furthermore, participants of that study were reported to be unaware of the influence of the priming when making their choices.

Applied to any large scale system, this becomes important for the purpose of determining user interest and providing more suitable/targeted information while reducing the visible intrusiveness of the system. As [11] reported, user feedback cannot be fully relied upon without the risk of fur-

ther intruding on user privacy by removing anonymity, and won't be given willingly by users in most cases anyways for fear of embarrassing themselves or even out of sheer apathy. Furthermore, since many users do not make any changes or customizations to their software except as almost a last recourse[27], users cannot be readily relied upon to indicate any changes in interests explicitly after the initial setup.

2.1.3 User environment as a context

Environmental psychology states that an individual's environment (such as weather, temperature and other such meteorological events) can have a significant effect on the mood and inclination of an individual[12][13][19]. In [36], Tai and Fung explored the influences of environmental cues in stimulating a user's pleasure and thus willingness to make purchases. Furthermore, the experience and mood of the user has a direct correlation, as seen in [35], in the willingness of the user to become receptive to advertisements and to carry through and even shop, indicating the importance of being able to both ascertain and keep track of a user's environmental data.

2.2 Related Works

2.2.1 Online advertisement

A lot of research has been done in the field of online advertisement for web pages, ranging from banners to video advertisements. As Burke et al[10] indicated, currently one of the largest obstacles to efficient online advertisement is the disruptive and often irrelevant nature of the advertisements.

This is especially true for video advertisements and to a lesser extent, text-based advertisements with frequently changing content. Both are considered to be even more intrusive due to their dynamic nature.[10]

In their research of scanpath theory, Josephson and Holmes[33] found that locations where audiences/users are accustomed to treating as 'advertisement spots' are afforded a different attention span than others areas of the screen. An example of this is the top of the screen. As [10] have shown, under certain page layouts, advertisement banners placed there are more memorable. Yet even taking advantage of this fact, several issues still hinder the effectiveness of online advertisements. Namely, they are easily forgotten[10]and they increase a user's perceived workload while hindering attention[21].

2.2.2 Psychologically Targeted Persuasive Advertising

In [33], Saari et al. explored the viability for a framework in "inducing desired emotion and attention related status." [33], the idea being that what the users experience will affect their subsequent willingness or inclination towards other behaviours. Their work pushed for the idea of the use of context, placing advertisements at 'well recognized' onscreen locations and even inducing endearment of the user towards the system in question, but did not go into detail on how they plan on doing that. While they briefly touch on many of the ideas mentioned in this thesis, their goal is a much more general framework than ours, which targets videos since they allow for a much greater level of granularity. Furthermore, their proposed system actively aims to impose emotions on a user, while ours simply offers the

ability to better gauge user state.

2.2.3 Collaborative and Content-Based Recommendation Systems

Recommendation systems provide users with suggestions on items that might be of interest to them. They typically employ two types of filtering methods in order to determine user preference: collaborative filtering[20] and content-based filtering[22]. Collaborative filtering offers recommendations to users based on the selections of other users with similar profiles/interests, and therefore assumed to have similar taste. Content filtering on the other hand provides recommendations based on the intrinsic properties of the item selected.

The most indicative use of collaborative filtering can be seen in commercial sites such as [9], [8], [4] and [7]. These systems make use of information such as user history and user feedback ratings in order to build up user profiles. These profiles are then compared [20][22], offering a user selections made by other users. Problems commonly associated with this filtering technique include the need for rating information from other users[22] in order to provide recommendations, a problem with the accuracy of matches if there is insufficient overlap between profiles[20], requiring user feedback and ignoring the potentially useful properties of the content being selected[37].

Content-based recommendations, on the other hand, make extensive use of the attributes of the content in order to make suggestions[30]. Attributes and keywords associated with the content are given strength ratings, the recommendations made based on matching said ratings through text

categorization.[29] However, current content-based recommendation systems can suffer from the problem of being unable to classify or define a user's profile[22] or give accurate predictions on possible future user interests[22].

Many video-on-demand(VoD) systems[6][4] make use of a hybrid approach, using the movies or channels that the user subscribes to as well as user provided ratings in order to decide what kind of recommendations are suitable for the user while matching their selections against other profiles. Others, such as [14], assigns categories to advertisements and entire videos in order to make matches. As well, they make use of data from explicit user interactions with past advertisements in order to determine which advertisements to push.

In this regard, the main difference between our framework and the others is that we don't particularly care about what the user states as their desired channel or preferences or which movie they select. Instead we base our advertisement suggestions on what the user actually watches. For example, if a user only watches a particular segment of a film repeatedly(say, a particularly good chase scene), then we only extract user preference information from that particular segment. This yields significantly different data than if we were to use traditional methods of profiling based on movie selection[8] or even channel registered[4], which is even more problematic since channel contents are not homogenous(i.e shows of Fox aren't all the same). Furthermore, considering that similar content can be found on different and possibly philosophically opposite channels, the unreliability of using channel selection as a criteria for user profiling becomes even more evident.

Chapter 3

System Design

In this chapter, we present the design of our advertisement/recommendation framework. By refining the granularity of existing context-based advertisement strategies as well as through the incorporation of the user's physical environment as a deciding factor, we aim to improve the quality of recommendations made.

3.1 Introduction

The targeted application of our framework is over an existing system which allows for collection of relevant or significant data concerning a user's preferences. Although the issue of privacy is certainly a concern, since we only require the system to retain locally what it has accessed, we feel it is acceptable. Furthermore, the underlying system in question should support some method of indicating the nature of the content currently being accessed by the user. While these criteria match that of most existing on-line browsing applications, it is especially true for the increasingly prevalent field of video-on-demand, which we will be examining in detail in the next chapter. From hereon, we will use the terms media and video interchangeably to refer to the content within the system that the user is interacting with.

Making use of the knowledge of what users have recently experienced, the framework determines the specific type of advertisement that the user will be most agreeable to, and subsequently retrieves it from the network. Since the design of our framework is independent of its underlying layers, the nature of the network it is operating over is not a limiting factor as long as it provides some method for storing categorized data. We provide, however, some insight into possible design considerations for the underlying network based on whether it is a central server, hybrid(i.e super-peers) or pure p2p.

3.2 System Architecture

Our framework can be divided into three chief components: the advertisement decision component, the advertisement storage/retrieval component, and the advertisement presentation component.

3.2.1 Advertisement Decision

The advertisement decision component refers to the aspect of the framework responsible for determining what type of advertisements would be most suitable/well-received by the user at any given time. This recommendation is done through an amalgamation of the user's past/usual preferences, the immediate stimulus/stimuli they are being exposed to, the nature of the content they are being exposed to and the physical environment that the user is in.(See Figure 3.1)

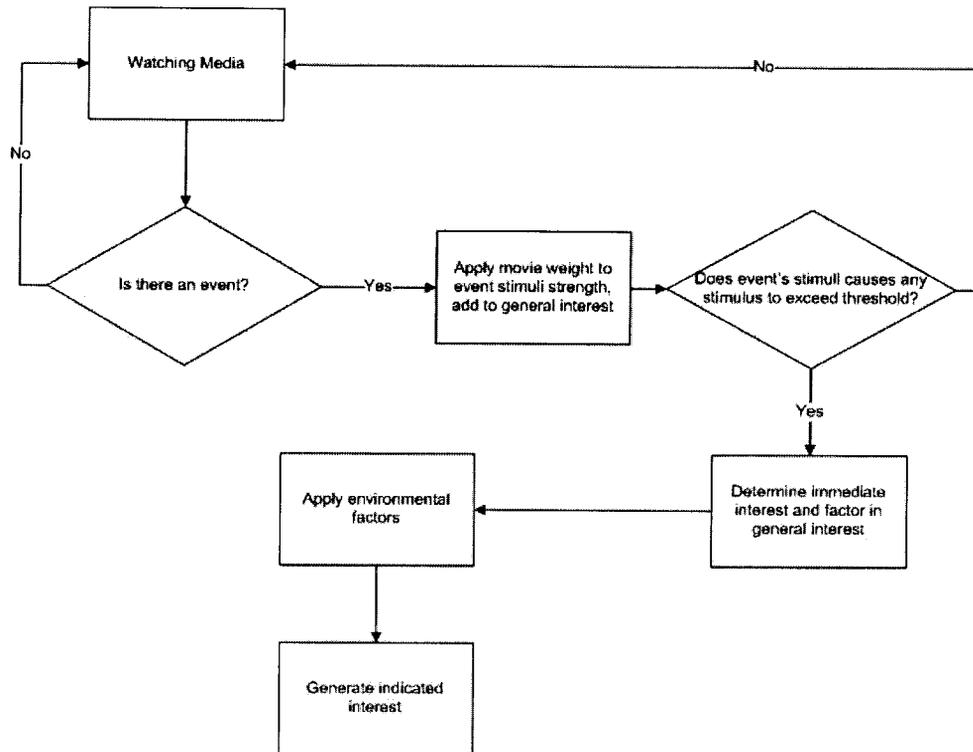


Figure 3.1: Flowchart for determining user's indicated interest

Interest and Stimuli: In defining a user's interest, we introduce the concept of stimuli, which represents the various influencing factors that could affect a user's taste and what type of advertisements they become interested or disinterested in. As [34] and [39] indicated, users are susceptible to the effects of contextual priming for both visual and verbal cues, which these stimuli represent. In most cases, we represent user interest in the form of a set of stimulus/value pair, with the value representing the current strength of the stimulus' effect on the user.

(Stimulus, Value)	(Stimulus, Value)
-------------------	-----	-----	-------------------

Table 3.1: Representation of User Interest

We further divide user interest into three categories:

Immediate Interest: In our framework, each event a user experiences is associated with some sort of information that indicate both the type of stimulus/stimuli that the event affect, as well as the strength of said stimuli. It is through the use of these values that we eventually decide what the user's immediate interest is.

As such, the immediate interest of the user represents what type of stimulus the user is most aroused by at any given time during their use of the application. This interest is defined when one or more stimuli's value, which continuously changes as the user experiences events that relates to those interests, exceeds a certain given threshold. This represents when the user has been influenced by their experiences to a degree sufficient to become susceptible to a specific type of information/advertisement[18]. The threshold can be anything, ranging from an arbitrary value to some product of the user's past history as well as the nature of the content they are interacting with, dependent on the underlying implementation.

In cases where multiple stimuli are being triggered simultaneously, the system will either select advertisements that satisfy all of the triggered stimuli or decide which one has priority or is relatively more seductive. Though this is dependent on the implementation of the framework, it should again be based, at least in part, on the individual user's past preferences, which

we refer to as the user's general interest.

General Interest: A user's general interest represents the usual preferences and priorities of the user. As mentioned earlier, the structure representing general interest contains values for all possible stimuli, as well as the associated strength for each. In addition, however, it also contains the weight for each stimuli.

(Stimulus, Value, Weight)	(Stimulus, Value, Weight)
---------------------------	-----	-----	---------------------------

Table 3.2: Representation of General User Interest

While the value represents the current level at which a certain stimulus is affecting a user, the weight represents the general susceptibility of the user to that stimulus. Both of these attributes will increase or decrease based on the user's activities(i.e. through the type of content they interact with) as well as other circumstantial factors, such as time and weather(which we examine later). Consequently, the general interest indicates the user's current state of being as well as being a representative of user history. As can be expected, the profile generated by this set of data will have a significant impact on which advertisements the system will seek for the user. These advertisements will all belong to the category of the user's indicated interest.

Indicated Interest: The indicated interest of the user is the criteria the system will use when it retrieves additional information from the network to display to the user. By taking into consideration the past preferences(general interest) of the user, the things that have affected the user the most in the

recent past(immediate interest), as well as the system's own environmental context, the system will derive the type of information the user is likely to want or will be intrigued by. This is expressed in the form of a query that the system sends to the network to retrieve an advertisement that satisfies what the user wants by matching advertisement category(or categories) and weights to that of the stimuli that have exceeded threshold.

The system will then, until the expiry of the interest, retrieve information from the network relevant to this indicated interest and provide it to the user. Expiry of an interest occurs when either a new interest is triggered by the system(hence replacing the current interest) or when the arousal levels for sufficient opposing triggers render the request for the currently indicated interest either nonsensical or no longer suitable. The user's general interest level will be affected by the satisfaction of the urges generated by the arousal of the stimuli through these advertisements, and the corresponding weights and values for the stimuli in question will again increase or decrease accordingly. For example, as a user experiences advertisements for a certain stimulus, the strength value for that stimulus will decrease to indicate a reduction in user appetite for that stimulus.

Environmental Factors: An often overlooked factor that plays an important role in the decisions made by a user is the physical environment in which they are using the system, and the effects of environmental psychology. As [37] and [36] have indicated, buyers are heavily influenced by the environmental cues available in the store, which affect their perceived level of pleasure and subsequently their buying tendencies. It stands to reason

then that the use of the computer's local environment, such as system time, local weather, barometric pressure, seasonal events and such should play a role in deciding what type of information would be better received by a user.

Furthermore, common sense points to many direct associations that can be made between environmental conditions and what a user might be interested in. For example, warm or hot weather would make the advertisement of parkas less attractive.

As such, the indicated interest of a user would be filtered by the environmental variables of the system. A number of ways to collect such information is viable, the least intrusive of which is for the user to input their location, though other possibilities such as traceroutes, whois, or even reverse DNS lookup are all viable. Once the general geographical location of a user is ascertained (and this will likely need to be done only once per session), it becomes trivial to retrieve the aforementioned environmental information and construct a rule-set.

3.2.2 Advertisement Storage and Retrieval

The advertisement storage and retrieval component refers to the aspect of the framework responsible for storing and retrieving advertisements back to the user. This component receives the recommendations made by the advertisement decision module, retrieves them from the network, and sends them back to the advertisement presentation component.

As mentioned previously, the nature of the network and the method of advertisement retrieval are closely connected. Regardless of what type of network our framework is applied over, however, certain common structures

will exist in each node.

Common Structures:

Advertisement Objects: Advertisement objects represent the actual advertisement offering products and/or services. Each object has the following information:

Name	Description
Activation Date	The date this advertisement becomes active. Preloading advertisement objects can allow for timely delivery of the advertisement information, and more importantly, to ensure availability.
Expiry Date	The date this advertisement expires. Useful for time-limited offers or seasonally relevant ads.
ID	Unique ID for advertisement.
Categories	The categories/stimuli that the advertisement belongs to, as well as the associated weight value for each stimulus present in the advertisement.

Table 3.3: Attributes for advertisement objects

Advertisement Storage: The advertisement storages are structures maintained in each node that manages advertisements for a particular type of stimulus. Each storage object contains unique advertisement objects, but one advertisement object can exist in multiple storages to represent advertisements that appeal to more than one category of interest, satisfying multiple stimuli. How the system determines which advertisement object

is inserted into which storage depends on the criteria one sets to consider an advertisement's weight value for a stimulus sufficient to qualify it for a category's storage. The advertisement storages are only used when advertisements for a specific stimulus is added(i.e. retrieved from the network due to a request from the advertisement decision component) or when the system receives a request for a locally stored advertisement from the network.

Each advertisement storage contains:

Name	Description
Stimulus	Type of stimulus that the storage is responsible for.
Advertisements	Contains the ID and file location of the advertisement objects stored in this node that is of the same stimulus type as the storage.

Table 3.4: Attributes for advertisement storages

Master Advertisement List: Each node contains a master advertisement list, which keeps track of all the advertisements stored locally and is responsible for managing the advertisement storages and objects in terms of removals. The master list contains:

Upon receiving a request for advertisement by the decision component, or when a new advertisement is inserted into the system(i.e. after being viewed), the master advertisement list goes through its expiry list and removes all the advertisements that have expiry times before the current time, making use of the category information found in the Advertisement List to ensure that deletes are cascaded to all relevant advertisement storages that

Name	Description
Expiry list	Contains a list of expiry dates and the advertisement ID that corresponds to each date, sorted by time.
Advertisement List	Contains a list of all existing advertisements(ids) in the local node, their file location, and the categories that they belong to.

Table 3.5: Attributes for master advertisement list

contain the advertisement.

Retrieval Models: Here, we will examine the changes we will need to make to system design for each of the more prevalent network models available(client/server, unstructured p2p and structured p2p(i.e. DHTs) systems) to accommodate our framework.

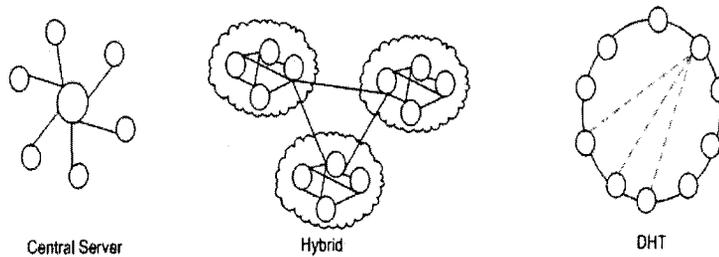


Figure 3.2: Types of networks

Central Server and Unstructured P2P: In a central server model, the server would contain a database of advertisements. Searching for an appropriate advertisement would then be a simple matter of submitting standard SQL queries to the server to match the weight/stimuli combination

desired. The use of a central server would allow for a very fine fidelity of matching to be done for the advertisements to be retrieved. After the appropriate advertisement is found, the server returns the file. Centralized p2p networks would operate in a similar manner except it would instead just provide the querying node with a list of nodes that possess an appropriate advertisement and its corresponding advertisement id.

For unstructured p2p systems where superpeers/seedpeers are deployed, they can be used as a means of processing the queries while the advertisement objects will be spread across the system based on user activity and existing clusters/domains. Beyond that, the difference between the central server and hybrid model is minimal. However, one will need to have some way of ensuring that a search will be complete in terms of advertisement searching/availability since there is no central system to rely on.

Ensuring that the network has at least one copy of each valid advertisement at any given time by making use of superpeers as seedpeers is one option. By forming a backbone/overlay between the superpeers[25], clusters can communicate easily and the overlays be given the responsibility of handling the different categories.

Structured P2P: Over structured p2p networks(i.e those that make use of DHTs)[32], there are several challenges in organizing the network to support our framework, since we search for advertisements based on their attributes instead of their filename, and these attributes are not unique. Because each advertisement often possesses more than one stimulus/weight pair, a combinatorial explosion of entries occur if we try to cater to all

possible categories via DHT alone[15]

One viable option for overcoming this hurdle is through the use of ontological directory indexing[23]. By providing each node in the system with an associated directory path that is also inserted into the DHT in addition to maintaining filename hashes, queries for a specific type of advertisement can be channeled down the appropriate directory path to the necessary domain to retrieve a suitable advertisement.[23] In order to account for advertisement objects falling under more than one category/domain, intermediate nodes can route a request to destination nodes that represent any of the domains the requested advertisement belongs to. Since we will be distributing the advertisement objects based on filename hashes initially, each node in the system will likely contain advertisements for multiple categories. With regards to the problem that would be caused by heterogenous advertisement popularity and the overhead this could impose, dynamic load balancing strategies, such as those mentioned in [24], are viable options.

Beyond the underlying network and security considerations, however, advertisement addition and removal over DHT is simple.

Addition involves using a simple set of rules to decide how an advertisement's stimuli weights qualify or disqualify it from a category, while removal would likely be based on a hard-state approach. Upon any node retrieving an advertisement, if it is found to have expired(via the expiry date information in the advertisement object), the retrieving node sends a message to the target node to request for a new advertisement of a similar type(which the target node should be able to retrieve on our behalf if we are employing ontological directory indexing). Since the advertisement's filename is now

known by the requesting node, it can notify all other nodes that possess the file via DHT lookup. Alternatively, the network could wait for each local node to remove the expired advertisement in their next storage/removal cycle.

3.2.3 Advertisement Presentation

The advertisement presentation component refers to the aspect of the framework responsible for presenting the retrieved data to the user. While a plethora of methods exist for displaying advertisement content to the user, considering that our target system is one that deals with video, an approach that does not impose a significant additional workload on the attention of the user would be ideal. For example, an initial simple text advertisement in the form of a slogan, at either the side or bottom of the video, is more likely to be accepted[10][28]. While it is decidedly less visually attractive, the workload for watching videos is already high enough without the additional perceived workload imposed by advertisements[10] being on-screen with the video content simultaneously. Of course, any audio advertisements would be completely inadvisable for obvious reasons.

In order to account for user fatigue and advertisement forgetfulness[10], however, the system would allow users to mark up advertisements that they find interesting, which will be displayed at the next system or user designated interval, using this as a form of feedback regarding the accuracy of the recommendations. A possible incentive for users to give feedback in this form is to provide them with an equal number of advertisements per break regardless of whether they provided feedback or not. Since they will be

spending a roughly equal amount of time viewing advertisements anyways, they might as well have a say in what kind of advertisements they will get. By relying on the contents of the video as well as the user's inclinations, we address the issue raised in [22] regarding the inherent problem of content-based recommendations failing to predict new user interest.

It is worth noting that the incorporation of the aforementioned 'commercial breaks' where queued advertisements are to be displayed provides an interesting design issue regarding video segmentation. Since the content flow of the video does not necessarily (and in fact, likely would not) coincide with how the media file itself has been segmented, injecting the queued advertisements between segments is not a viable option. We return to this problem in Chapter 6 when we examine future avenues of refinement for our framework.

Chapter 4

Implementation Analysis

In this chapter, we explore the viability of our framework over an existing system, using the BitVampire[25] concept as an example of how our framework can be applied. Being a peer-to-peer video-on-demand architecture, it is uniquely suited for showcasing our proposed framework and highlighting certain design issues. We begin with a brief explanation on the concept for the relevant parts of BitVampire and its underlying search infrastructure, COOLSearch. Then, we detail the modifications necessary to apply our framework.

4.1 Introduction

BitVampire[25] is an architecture designed for on-demand video over heterogeneous p2p networks. The premise of BitVampire is to divide a video file into multiple smaller, chronologically and locally ordered segments, which are requested in sequence by a user when they wish to watch the video. By segmenting the video into multiple files, and then leveraging the fact that multiple peers in the system likely possess the desired segments, a higher download speed can be achieved without overburdening any particular source. These segments are distributed throughout the network during

the initial publishing of the video, and then, dependent on the caching policy of the video and the local node in question, retained by users who have downloaded that segment.

Although the BitVampire technology is independent of the underlying search or network model, it is often associated with the category overlay search(COOLSearch[38]) infrastructure, which is based on a modified 'Super Peer' overlay architecture. COOLSearch divides the network into multiple clusters, each maintained by a designated core node(typically, the node that formed the cluster). Within each cluster is a representative of an existing semantic category, which is referred to as an agent node. As such, each node can act as an agent node for 0 or more categories. Agent nodes of similar categories are connected by what is referred to as logical links such that there are n overlays in the network, where n is the number of categories denoted.

For example, any object published into the system will belong to a category x , and could be distributed to any number of clusters. The agent node for each category in the corresponding clusters are responsible for keeping track of objects in its cluster that belongs to its category. As such, whenever an object needs to be found in the network, the querying node only needs to locate the agent node responsible for that category within its own local cluster, which will then search through its category overlay for the desired object and reply with the result.

4.2 Clip and Video Metadata

In order to support the ability to mark up event occurrences in a media segment, we need some way of associating a segment's metadata with its corresponding file object. BitVampire uses the resource.BitVampire.MediaSegmentInfo class to represent segmented media objects and some of their properties, so we insert the information necessary there by adding a new metadata class object, MediaMetadata(See Table 4.1).

Name	Description
StartTime,EndTime - Time	Represents the time interval that this segment occupies relative to the entire media file. This is necessary to allow for the times represented in the list of events to be read appropriately.
EventsList - Hashtable<Time,StimEvent>	Represents the events that occur during the clip and the time interval during which they occur. The time indicates the events relative to the entire media file rather than within the segment.

Table 4.1: resource.BitVampire.MediaMetadata.java

Events: Events(See Table 4.2), as indicated in previous chapters, refer to specific time intervals within (in the case of VoD systems) a video where certain stimuli are triggered by on-screen content. Depending on the granularity with which we mark up segments, an event could be something as mundane as a landscape scene(i.e snow-covered hills making people want to ski) to particularly dramatic or thrilling on-screen moments. A segment is composed of events, which in turn are composed of sets of stimuli of varying

strengths to denote their respective influence.

Name	Description
StimuliList - List<(Stimulus, Strength- int)>	Represents the various stimuli that a user that watches this event will be affected by, as well as their associated strength.
Duration - Time	Indicates the duration of the event. Depending upon the granularity with which events are marked up, duration could be useful in capturing users who experience only a part of an event.

Table 4.2: advertisement.StimEvent.java

Movie theme: The media that the individual segments are constituent of, which in the system is represented by a resource.MediaResource object, also needs to be modified to support stimuli effects. As we indicated in Chapter 3, the overall theme of a media would heavily affect the type of content found within, and therefore the impact of any individual stimulus. In order to account for this, we provide an overall modifier(See Table 4.3) whose effects will be applied to all events in a movie's segments.

Name	Description
Modifier - Hashtable <Stimulus, Weight - double>	Weight modifiers are necessary since a movie might, for example, be overloaded with a certain type of stimulus. If a normal/equivalent weight is maintained for all the stimuli, we might always only generate a certain trigger, which while accurate in terms of on-screen content, would quickly lead to fatigue on the part of the user.

Table 4.3: Movie modifier structure

4.3 Advertisement Components

4.3.1 Advertisement Decision

The advertisement decision component, which represents both the user's profile and the decision making process within the system, is responsible for generating the recommendation that the user is most likely to be agreeable to. (See Figure 4.1)

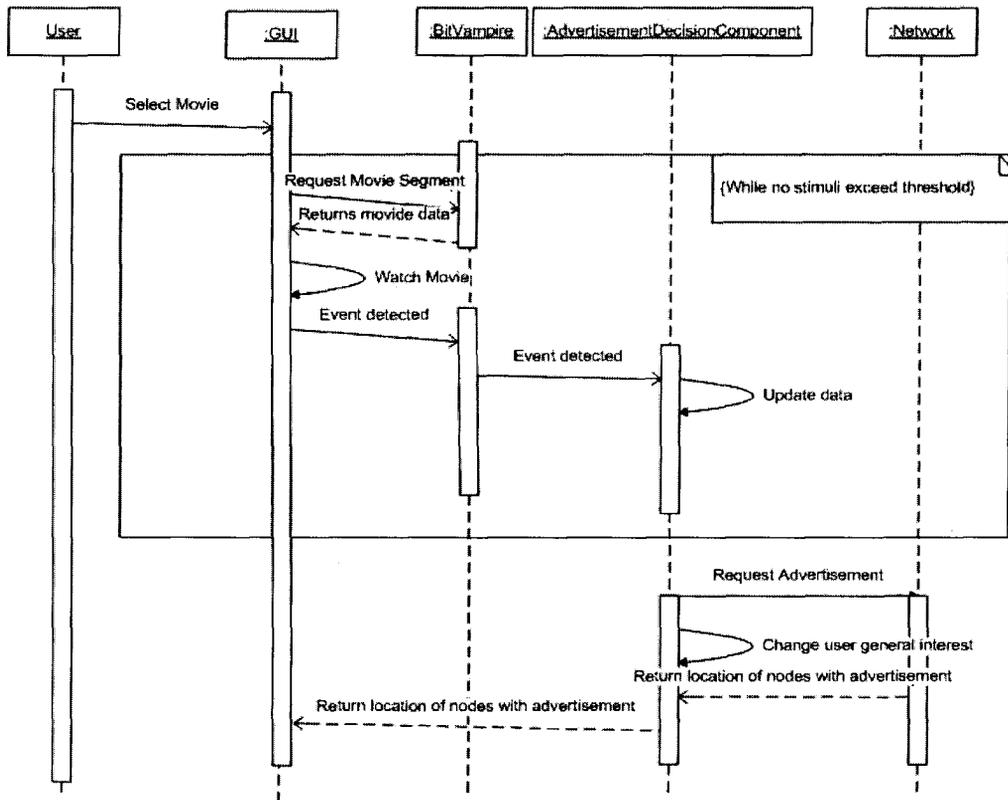


Figure 4.1: Overview of advertisement decision system

General Interest: General interest represents the profile of the user, and is a reflection of user history. Since `kernel.LocalNode` represents the local node in BitVampire and is always instantiated as one of the first system operations, it would be an ideal place to insert our singleton `GeneralInterest` object(See Table 4.4).

Name	Description
Threshold - int	Threshold is the value that any associated stimulus strength has to match or exceed in order for an advertisement request to be sent for said stimulus(uli). In our case, it is arbitrary.
CurrentExcitement - Hashtable<Stimulus, (weight - double, strength - int)>	This represents the current state of the user. Stimulus is paired with the weight(which represents how susceptible the user is to said stimulus) and the strength(how affected are they by the stimulus currently).

Table 4.4: advertisement.Decision.GeneralInterest.java

Immediate Interest: Immediate interest represents the stimulus/stimuli that a user is currently most affected by. This component acts as a listener, waiting until the user is sufficiently excited and then initializing the process of requesting an advertisement. A user's excitement level changes according to which events they interact with, as well as the frequency of interaction. As we indicated in Chapter 3, environmental considerations also come into play in determining user susceptibility to any particular stimulus. We now detail the structures and operational mechanics for detecting immediate interest and a simple representation of environmental effects.

Changing excitement level: Events are continuously detected by the system during the viewing of a video segment, their corresponding stimuli effects altering the excitement values in the CurrentExcitement records(See Table 4.4). When one or more stimulus exceeds the designated threshold, an immediate interest is generated by the system based on which stimulus triggered it.[See Appendix A - Threshold Trigger Detection] The immediate interest is then used in deciding the indicated interest.

Environmental factoring: Environmental factoring represents the influences that weather, barometric changes and other such conditions have on the mood and inclination of a user and subsequently what they are more or less susceptible to. Since the creation of a rule-set or analysis mechanism that can understand the significant of meteorological data and how they translate to human shopping moods is beyond the scope of this thesis, we arbitrarily represent environmental factoring as simply a hashtable with stimulus as key and weight as the value for each environmental factor. This weight represent the dampening or intensifying effect the environmental factor has on the respective stimulus. A representation of a user's full environment would then be a composite of environmental factors used to indicate the combined effects. (See Table 4.5)

Name	Description
EnvironmentalModifier - Hashtable <Environmental Factor, Hashtable<Stimulus, Weight - double>>	Applied to general and immediate interest to find indicated interest.

Table 4.5: advertisement.Decision.EnvironmentalFactor.java

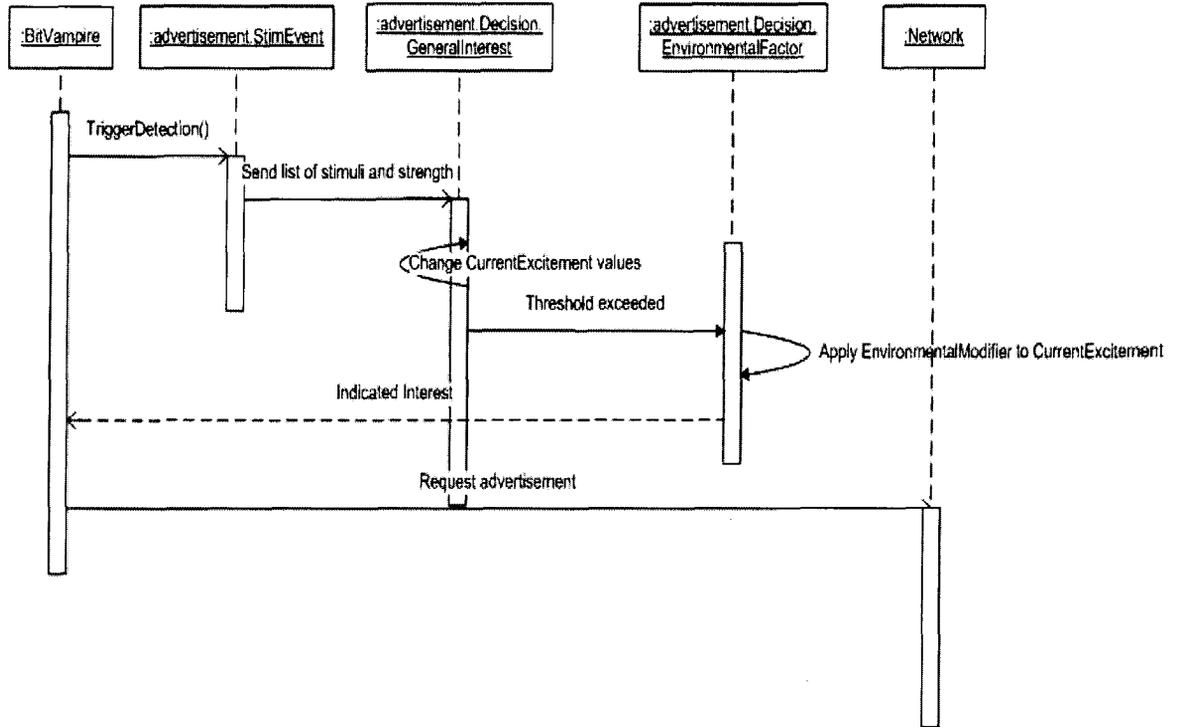


Figure 4.2: Sequence diagram of the generating indicated interest

Indicated Interest: Indicated interest represents the request sent by the local node to the network when the strength of any stimulus exceeds its local designated threshold value. For the purpose of our prototype, the indicated interest will be the stimulus that has the highest strength rating after environmental factors have been applied to the general interest’s CurrentExcitement table upon detection that a stimulus has exceeded threshold. (See Figure 4.2)

Name	Description
ID - int	Unique number to represent the advertisement in the system. Since each category employs its own agent node and overlay, this can be ensured.
ActivationDate,Expiry - Date	Both activation and expiry dates will be checked to ensure that the advertisement should be displayed to the user.
Description - String	Text description of advertisement, such as slogan, etc.
StimuliList - List<Stimuli, Weight- double>	This is similar to the event list found in clip metadata, except here we need to represent how much of each component stimulus an advertisement has(relatively).

Table 4.6: advertisement.Storage.AdvertisementResource

4.3.2 Advertisement Storage and Retrieval

Advertisement Objects: As seen in Chapter 3, advertisement objects refer to the description data for advertisements that can be presented to the user. In the case of BitVampire, we will be treating the actual advertisements as we do any normal media object(so we can represent them using the existing resource.MediaResources), publishing and distributing them based on their stimuli/categories. In order to represent the relevant advertisement information for these files, we create a new class, AdvertisementResource.(See Table 4.6)

Storage and Master Advertisement List: The advertisement storages(Table 4.7) and the master advertisement list(Table 4.8) objects are almost identical in implementation to their conceptual design as per Chapter 3, so we will only indicate their structures.

Name	Description
Stimulus - Stimulus	This value would be unique, since we each storage represents one type of stimulus.
Advertisements Hashtable<ID - int, Ad- vertisement - File>	Since IDs will be unique(see above), hashtables can be used.

Table 4.7: advertisement.Storage.StimulusStorage

Name	Description
ExpiryList - List <Time, ID - int>	Lists are used since multiple advertise- ments could have the same expiry time.
AdvertisementList Hashtable<ID - int, (Advertisement - File, List<Stimulus>>	We need to keep track of each advertise- ment's associate stimuli so that any re- movals will be appropriately cascaded.

Table 4.8: advertisement.Storage.MasterList

Advertisement Insertion: Advertisement insertion/retention is based on advertiser setting(whether they allow users to store the advertisements after viewing it, etc.). Since the insertion of the actual advertisement file is part of BitVampire's file maintenance function, which deals with storage space restrictions and local caching policy, we will only provide the pseudo-code for insertion with regards to the management of advertisement objects. [See Appendix A - Insert Local Advertisement]

Advertisement Removals:

Local: We remove advertisements upon expiry, and this is checked whenever an advertisement is requested from or added to the local node. We make use of the existing BitVampire media resource removal function in resource.ResourceDBManager to remove the actual advertisement object

(since we treat it as though it is any other 'clip' in the system), which will notify the corresponding agent node that we no longer possess that file.

Network: With regards to the issue of advertisement management across the entire network, while we try to alleviate message and bandwidth overheads by having local nodes check for advertisement expiry individually (versus flooding the network to announce when an advertisement expires), there are situations where we would conceivably want to remove or change the information about a specific advertisement, and have this change propagated throughout the network. To accommodate this, we modify BitVampire's existing communications package to allow for an advertisement's information to be over-written after the fact. Since the category overlay can be used to locate the nodes that contain the advertisement object in question, changing advertisement objects is just a matter of contacting the nodes and providing them with new information. Likewise, forcing an advertisement to be removed from the network would simply be equivalent to changing the expiry date to an arbitrarily early date so that at the next storage/removal cycle, the advertisement object in question, along with its associated file, are removed.

4.3.3 Advertisement Presentation:

When an appropriate advertisement object has been retrieved from the network, it and its associated media file are stored in the system. Using the text description contained in the media's metadata, this description (likely a slogan of sorts) is displayed, and the user has a choice to either dismiss it

Chapter 4. Implementation Analysis

or queue it up to be displayed later. If queued up to be displayed later, the advertisement media file will be one of the advertisements displayed during the next designated 'commercial break'.

Chapter 5

Evaluation

In this chapter, we evaluate the workload our proposed framework would impose on a given system. To do this, we examine the number of advertisement requests issued from the framework as we increase the granularity of detail with which it monitors user preferences and on-screen content. Specifically, we examine the effects of 3 different parameters on the number of advertisement requests generated by our system: N_s , the number of Stimuli(i.e. number of categories), F_e , the frequency of event markup in a media segment and Avg_{es} , the average number of stimuli affected by an event. As we indicated in previous chapters, since we do not have the necessary rule-sets to represent the real world, nor users who we have sufficiently profiled, we have created our own simulator to test our framework. As we will mention in our future works section, the effects of each stimulus is likely to bleed over and affect the strength or weight of other stimuli in the same event. For these experiments, however, we will be treating each stimulus as an independent trigger that does not impose any secondary or tertiary effects of other stimuli for the sake of clarity.

5.1 Simulation Setup

Environmental Pool: In our simulation, we generate 1,000 nodes, allocating each of them with random general interests (both weights and initial starting stimuli strength is randomized), to represent a sample group of users. We also denote 4 criteria used to derive a user's physical environment, which we term: weather, season, temperature, and barometric pressure. For each criteria, we generated 30 random sets of stimuli/weight pairs to represent the effect each of these criteria would have on a user, such that a total of 30^4 possible environments are possible. From this, we pick 100 to form our environment pool. Each node is then assigned an environment setting from this pool.

Movie Segment Pool: We also created a movie segment pool to represent the possible movies in the simulation. First, a movie modifier pool is generated to represent the effects the overall theme of a movie has on its constituent segments, assigning a modifier to each possible movie. Movie segments are created, each populated with events containing randomized stimuli and strength. These segments are then assigned a movie modifier to categorize them into different movies to form our pool.

We simulate an average session of video watching and examine the number of advertisement requests generated per user and how the changing of our 3 parameters will affect it. For the purpose of this simulation, we did not incorporate any deterioration to stimuli due to time, and only rely on events with negative stimuli strength to bring a user down from threshold.

This represents the worst case scenario and is useful to gauge the maximum stress the framework is likely to inflict. Each simulation is run 50 times and the results averaged.

5.1.1 Simulation Parameters

Unless otherwise stated, the following parameters will be used for all of our experimentation. For this simulation, we generated a total of 1000 movie segments that a user could be watching. The average start time for each of the 1,000 users is set between 0 to 25% of the total length of the simulation to represent random watch times. We set the granularity of events(F_e) to be 2 per segment(roughly equivalent to 25 seconds per event on average), with a total of 50 stimuli(N_s) being monitored by the system. Each event on average affects 20% of all stimuli(rounded up), with a 50% chance that a stimulus would have a negative effect[34] on the user, and the average strength of a stimulus in any event is 1% of the average threshold value. The length of the simulation is set for the time it takes 50 segments to be processed, equivalent to about 42 minutes of watching(which is about the average time of a normal television series episode, sans advertisements).

The first parameter we investigate is the number of stimuli(N_s) in the system. One of the most important factors to consider for the framework is the granularity of the type of advertisements it can support. This translates to the number of stimuli that needs to be represented in segment events and environmental effects and to be maintained in user profiles.

The second parameter we examine is the average frequency of events(F_e) generated by each segment. This allows us to gauge the level of detail that

can be afforded by the system in keeping track of on-screen events, and by extension, how closely the user's actual state is reflected by the user's profile in the system. Closely related to the effects of event frequency is the impact that the average number of stimuli an event is represented by ($Avges$), and what effects it will have on advertisement requests made, which is our third parameter. We examine this last variable in terms of its value relative to the total number of stimuli in the system.

5.2 Simulation Results

This section presents the results of our simulation and the conclusions we can draw from them regarding our proposed framework. We examine the effects our parameter changes have using the metric of the number of threshold breaks (and thus advertisement requests) made in the system per minute. In order to generate our results, 50 rounds of simulations were run, each representing 50 minutes of activity in an network of 1000 nodes, with each user/node having a different starting time and thus different viewing length. The results are presented as an average of all runs.

Stimuli Granularity: In this set of experiments, we examine the effects that changes to the total number of stimuli (N_s) present in the system would have to the number of messages our framework would generate. We set the initial granularity of N_s to 2, as per [11]'s classification of emotional vs. non-emotional types of advertisements. Then, we change N_s to 25, 50, 75 and then 100, maintaining the ratio of total possible stimuli to average number

of stimuli per event($Avges$) at 20%(rounded up).

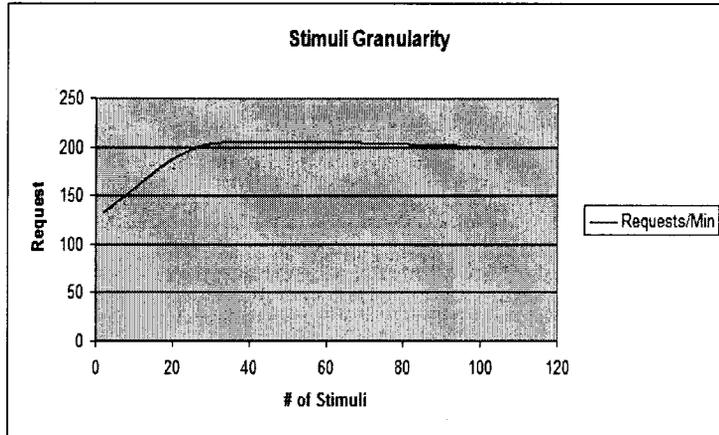


Figure 5.1: Average number of advertisements per minute

We can see from Figure 5.1, which indicates the average number of advertisement requests generated by the system per minute, that the number of messages generated as N_s increased rises sharply, but quickly plateaus, showing only slight variation in numbers of advertisement requests issued as N_s continues to increase. We attribute this to the fact that as the number of stimuli being represented and thus accounted for increases, the strength assigned to each stimulus in an event decreases to reflect this increased granularity, so each stimulus takes longer to exceed threshold.

Event frequency: In this set of experiments, we examine the effect that an increase in the frequency of events(F_e) being marked in a segment would have on the number of messages being generated. As before, we set $Avges$ to 20% of all possible stimuli we represent and examine the effects of changes to event frequency. We begin with an arbitrary value of F_e to 2 events per

segment (approx. 25 seconds per event), 3 (approx. 16.6 seconds per event), 4 (approx 12.5 seconds per event), 5 (approx. 10 seconds per event), and lastly 6 events per segment (approx. 8.3 seconds per event).

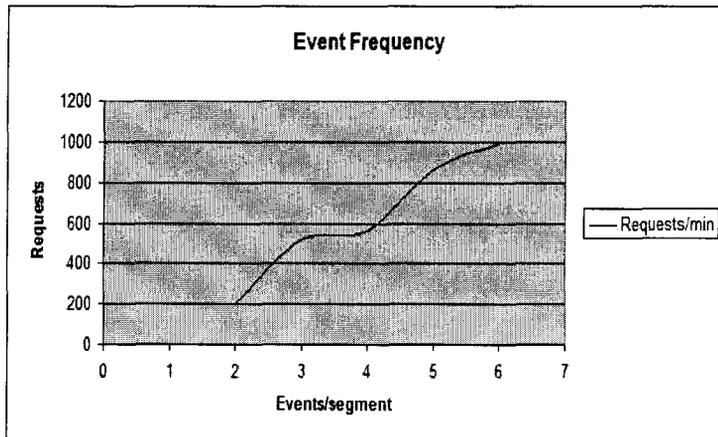


Figure 5.2: Average number of advertisements per minute

From Figure 5.2, we see that, as expected, increasing F_e in a segment would lead to an increase in the advertisement requests made by the nodes in the network. The roughly linear increase in requests per minute is caused by the additional stimuli being factored in. Since we are using only a 20% stimuli incorporation, stimuli collision was not a factor.

Number of Stimuli per event: In this last set of experiments, we look into how changes in the average number of stimuli ($Avge_s$) affected by an event has on advertisement request output. The difference between this set of experiments and the first is that in this case, we are examining the impact that changes in the percentage of represented stimuli out of all possible ones, instead of the number of possible ones.

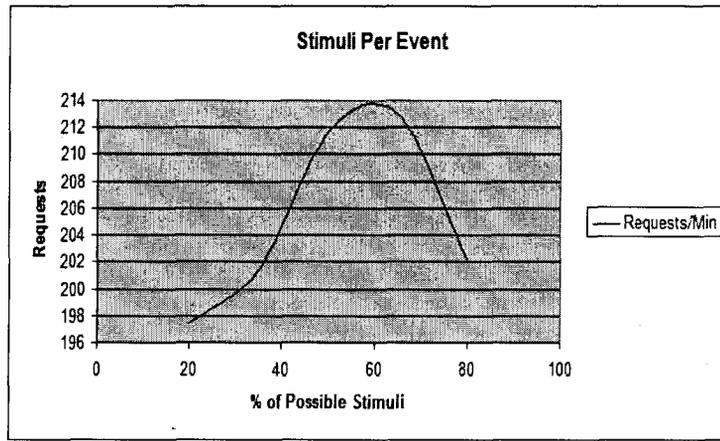


Figure 5.3: Average number of advertisements per minute

We examine the incurred message load when we change Avg_{es} to represent 20%, then 35%, 50%, 65% and lastly 80% of all possible stimuli in the system. The results are shown in Figure 5.3. While there is again the initial increase, we actually see a clear deterioration in the number of requests generated by the system as Avg_{es} gets closer and closer to the total number of stimuli in the system. This is again due to the collision factor seen previously since as more and more stimuli are represented, they invariably start canceling each other out. In this case, the cancelation is more dramatic than what we saw in Figure 5.1 since we are basically reducing the probability that each event's stimuli will not have an effect on that of previous events as opposed to just increasing the stimuli pool.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

While there has been a lot of work done in both the commercial sector and academia regarding recommendation systems and ways of gauging and predicting user interest, they all suffer from an oversight in the granularity with which they perceive user activity. With the emergence of the internet and video-on-demand systems, it is now viable to know exactly what a user is experiencing on-screen at any given time as well as the kind of user watching it. With the ability to break video content down into their constituent events, and these events into the constituent stimuli that they excite in a user, there is no reason why this information should not be used to help determine what type of advertisements a user will be most susceptible to at any given time.

In this thesis, we introduce SUE, a new framework for content-based advertisements by taking advantage of the system's knowledge in both user on-screen experiences and off-screen environment. By allowing the system to keep track of user interest via their activity and the type of stimuli they have been exposed to in both the recent and distant past, a profile can be built that determines a user's susceptibility to a particular stimulus.

Coupled with the use of meteorological data on the user's general location, which is easily accessible, it allows us to determine and provide targeted advertisements that will be received much more saliently.

In order to prove the viability of our proposed framework, we chose an existing p2p video on demand system, the BitVampire project, and examined in detail the design changes necessary to incorporate our framework into it. To evaluate our design, we simulated the workload our framework would impose on a randomly generated rule-set. We observed that as the number of stimuli the framework needs to keep track of, and thus the granularity with which we observe user behaviour, increases, the number of recommendations being made by the framework begin to either plateau or even decrease due to stimuli/event collisions. It is worth noting that this decrease in recommendations is not necessarily a bad thing, since we have no way of gauging the accuracy of the recommendations made in terms of how closely they appeal to the user's actual interests.

6.2 Future Work

There are several areas in the development of the system that deserve further exploration. The most evident is how the retrieved data is to be presented to the user. Obviously, the nature and strength of the retrieved advertisement, as well as the current on-screen data, will be factors in how the data should be presented to the user (i.e. We do not want to interrupt an action sequence with a pop-up). This issue of timing, which we touched on briefly in Chapter 3, is also constrained by the need to deliver the targeted advertise-

ments before the indicated interest expires. A viable avenue is the addition of even more metadata to mark time intervals within segments where it is more acceptable from the user's perspective for 'commercial breaks' to occur. Not only would this go to alleviate user annoyance, but it could also be incorporated into a business model where the frequency and timing of advertisements is dependent on the type of user.

The data analyzing component is also another aspect of the project that would benefit from more extensive work. While modular in nature, the application of the framework into any conceivable system will require the ability to decide what kind of things the user will be most receptive to being exposed to. Our prototype only composed of a simple weight and value based decision matrix, and environmental considerations are made only arbitrarily. However, a stimulus likely has secondary effects that would bleed over to affect the strength value of other stimuli(either intensifying or dampening it, in the case of conflicting stimuli). While there would no doubt be variations in the strength of the influences caused by such factors, the construction of a standard table of influences in relation to environmental variables(i.e general geographical location, time of day of access, season, weather, etc.) would be beneficial, with the applications of profiles and add-ons over this standard template used depending on the target audience of the system. Though there is certainly sufficient literature on the subject, the often conflicting results could be a problem.

An interesting though likely controversial topic is in how to ensure that the user will receive or perceive the events in a video to the degree that we intended. How would one factor in the mood of the viewer beforehand? Is

there anything that could be done to 'prep' the user/viewer to becoming more sensitive to a certain stimuli in order to make up for not experiencing all the previous content? While contextual priming is useful for the actual video content, is the use of it in, say, the type of GUI to display to a user in order to get them into the right mood a viable option?

Another area of exploration that is readily available and more functionally applicable is in content searching and classification. Traditional systems offer searching and matching based on cast, genre, etc. However, by making use of the existing event stimuli lists that are already present in the metadata for each segment of a movie, we can easily allow for users to search for specific content(i.e a specific romance scene) or even emotional experiences through a database of movies.

While local searching is relatively trivial since we already have an existing list of events that we can retain for any segment we have viewed, network topology again comes into play, as we saw in Chapter 3, when we want to look for events globally. In the case of structured p2p networks, these events could easily be inserted into an existing DHT system, but then the cost incurred in removals, or even common searching is likely to become a significant factor, even more so than the advertisement storage issues seen early. While categorization could be done based on stimuli presence, the mere fact that there will be so many events suggests that a simpler solution would probably be to employ a centralized database. Furthermore, since this search capability is not a critical system component, the problems likely to arise from single points of failures or bottlenecks are offset by the drastically reduced amount of bookkeeping.

Chapter 6. Conclusion and Future Work

Last but not least, once a set of rules for environmental and stimuli effects have been defined, extensive user studies would be desirable in providing the theoretical viability of this framework. Our evaluations can unfortunately be limited to only system performance until then.

Bibliography

- [1] Azureus - azureus.sourceforge.net.
- [2] BitComet - www.bitcomet.com.
- [3] Gnutella - www.gnutella.com.
- [4] Joost - www.joost.com.
- [5] Hulu - www.hulu.com.
- [6] Babelgum - www.babelgum.com.
- [7] Amazon - www.amazon.ca.
- [8] NetFlix - www.netflix.com.
- [9] eBay - www.ebay.ca.
- [10] M. Burke, A. Hornof, E. Nilsen, and N. Gorman. High-cost banner blindness: Ads increase perceived workload, hinder visual search, and are forgotten. *Transactions on computer-human interaction*, 12(4):423, 2005.
- [11] Jenna Burrell and Geri K. Gay. Collectively defining context in a mobile, networked computing environment. In *CHI '01: CHI '01 extended*

Bibliography

- abstracts on Human factors in computing systems*, pages 231–232, New York, NY, USA, 2001. ACM.
- [12] L. P. Chiu. Do weather, day of the week, and address affect the rate of attempted suicide in hong kong? *Social psychiatry and psychiatric epidemiology*, 23(4):229, 1988.
- [13] E. A. Deisenhammer. Weather and suicide: the present state of knowledge on the association of meteorological factors with suicidal behaviour. *Acta psychiatrica Scandinavica*, 108(6):402, 2003.
- [14] Moonka; Rajas ; et al. Using viewing signals in targeted video advertising, March 2008. United States Patent Application 20,080,066,107.
- [15] L. Garcés-Erice, P. A. Felber, E. W. Biersack, G. Urvoy-Keller, and K. W. Ross. Data indexing in peer-to-peer dht networks. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 200–208, Washington, DC, USA, 2004. IEEE Computer Society.
- [16] M. E. Goldberg and G. J. Gorn. Happy and sad tv programs: How they affect reactions to commercials. *The Journal of consumer research*, 14(3):387, 1987.
- [17] P. M. Herr. Priming price: Prior knowledge and context effects. *The Journal of consumer research*, 16(1):67, 1989.

Bibliography

- [18] Wim Janssens and Patrick de Pelsmacker. Advertising for new and existing brands: The impact of media context and type of advertisement. *Journal of Marketing Communications*, 11(2):113, 2005.
- [19] G. Jessen, P. Steffensen, and BF Jensen. Seasons and meteorological factors in suicidal behaviour. *Archives of suicide research*, 4(3):263, 1998.
- [20] Xin Jin, Yanzan Zhou, and Bamshad Mobasher. A maximum entropy web recommendation system: combining collaborative and content features. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 612–617, New York, NY, USA, 2005. ACM.
- [21] Sheree Josephson and Michael E. Holmes. Visual attention to repeated internet images: testing the scanpath theory on the world wide web. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 43–49, New York, NY, USA, 2002. ACM.
- [22] Junzo Kamahara, Tomofumi Asakawa, Shinji Shimojo, and Hideo Miyahara. A community-based recommendation system to reveal unexpected interests. In *MMM '05: Proceedings of the 11th International Multimedia Modelling Conference*, pages 433–438, Washington, DC, USA, 2005. IEEE Computer Society.
- [23] Juan Li. *Semantics-Based Resource Discovery in Global-Scale Grids*. PhD thesis, Department of Computer Science, The University of British Columbia, 2008.

Bibliography

- [24] Juan Li, Billy Cheung, and Son Vuong. A scheme for balancing heterogeneous request load in dht-based p2p systems. In *Proceedings of the Fourth International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2007)*, August 2007.
- [25] Xin Liu. Bitvampire: A cost-effective architecture for on-demand media streaming in heterogeneous p2p networks. Master's thesis, Department of Computer Science, The University of British Columbia, 2005.
- [26] D. J. MacInnis and B. J. Jaworski. Information processing from advertisements: Toward an integrative framework. *The Journal of marketing*, 53(4):1, 1989.
- [27] Wendy E. Mackay. Triggers and barriers to customizing software. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160, New York, NY, USA, 1991. ACM.
- [28] Sascha Mahlke and Manfred Thüring. Studying antecedents of emotional experiences in interactive contexts. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 915–918, New York, NY, USA, 2007. ACM.
- [29] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations, 2002.
- [30] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, New York, NY, USA, 2000. ACM.

Bibliography

- [31] SD Perry, SA Jenzowsky, CM King, H. Yi, JB Hester, and J. Gartschlaeger. Using humorous programs as a vehicle for humorous commercials. *Journal of Communication*, 47(1):20–39, 1997.
- [32] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.
- [33] Timo Saari, Niklas Ravaja, Jari Laarni, Marko Turpeinen, and Kari Kallinen. Psychologically targeted persuasive advertising and product information in e-commerce. In *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, pages 245–254, New York, NY, USA, 2004. ACM.
- [34] Bernd Schmitt. Contextual priming of visual information in advertisements. *Psychology marketing*, 11(1):1, 1994.
- [35] William Swinyard. The effects of mood, involvement, and quality of store experience on shopping intentions. *The Journal of consumer research*, 20(2):271, 1993.
- [36] Susan H. C. Tai and Agnes M. C. Fung. Application of an environmental psychology model to in-store buying behaviour. *The International review of retail, distribution and consumer research*, 7(4):311, 1997.
- [37] Kirk L. Wakefield and Julie Baker. Excitement at the mall: Determinants and effects on shopping response. *Journal of retailing*, 74(4):515, 1998.

Bibliography

- [38] Jun Wang. Efficient content locating in dynamic peer-to-peer networks. Master's thesis, Department of Computer Science, The University of British Columbia, 2005.
- [39] Youjae Yi. The effects of contextual priming in print advertisements. *The Journal of consumer research*, 17(2):215, 1990.

Appendix A

Pseudo-Code

Algorithm 1 Threshold Trigger Detection

```
1: TriggerDetection()
2: while User is viewing some media segment  $M$  do
3:   Find the Event  $E$ , where  $E \in EventList_M \wedge T_{E.StartTime}$  is closest to
     current time  $T_{current} \wedge T_{E.StartTime} < T_{current}$ 
4:    $\backslash\backslash$ This is usually the first element, but need to check in case a new
5:    $\backslash\backslash$ segment got cued up while we were sleeping
6:   sleep for that time
7:   if  $M_{current} == M$  then
8:      $\backslash\backslash$ Check if current media is the same as the queued event's
9:     Retrieve all of  $E$ 's stimuli,  $S_0 \dots S_n$ , and their corresponding effects,
      $Effect_{s_0} \dots Effect_{s_n}$ 
10:    Apply it to the user's general interest values,  $V_{user}$ 
11:    if  $\exists x \in V_{user} \wedge x > \text{threshold}$  then
12:       $\backslash\backslash$ Check if anything exceeds threshold
13:      Request an indicated interest
14:    end if
15:  end if
16: end while
```

Algorithm 2 Insert Local Advertisement

```

    InsertAdvertisement(advertisement id  $id_a$ , advertisement object  $a$ )
2: if  $\neg(id_a \in \text{master list's advertisement list}, L_{\text{master}})$  then
    Create new advertisement object  $a_{\text{new}}$  such that  $a_{\text{new}} = a$ 
4:   Insert into  $L_{\text{master}}$ 
    Insert entry into correct position in expiry list,  $L_{\text{expiry}}$ 
6:   Add to each corresponding stimulus storage,  $Storage_{s_0} \dots Storage_{s_n}$  ob-
    ject  $a_{\text{new}}$ 
    return
8: end if
    if  $L_{\text{master}} \neq \text{null}$  then
10:  while  $L_{\text{expiry}} \neq \text{null} \wedge$  the first event in the expiry list's time,  $T_{L_{\text{expiry}}[0]}$ 
     $<$  current system time,  $T_{\text{current}}$  do
    //Look for expired advertisements
12:   Get first expired advertisement  $a_{\text{expired}}$ 
    if  $id_{\text{expired}} \in L_{\text{master}} \wedge$  advertisement's expiry date,  $T_a =$  date in
    list's advertisement,  $T_{\text{expired}}$  then
14:     Remove all entries of  $id_{\text{expired}}$  from  $Storage_{s_0} \dots Storage_{s_n}$ 
    Remove associated media object via ResourceDBManager
16:     //This notifies the network that we no longer have
    //this piece as well as removes it from local storage
18:     end if
    end while
20: end if

```
