Analyzing User Feedback of On-Line Communities

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von

Clemens Schefels
aus München

Frankfurt am Main (2013)
(D 30)

vom Fachbereich Informatik und Mathematik der
Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan: Prof. Dr. Thorsten Theobald
Gutachter: Prof. Dott.-Ing. Roberto V. Zicari
Prof. Dr. Miguel-Angel Sicilia

Datum der Disputation: 04. Juli 2013

Abstract

The economic success of the World Wide Web makes it a highly competitive environment for web businesses. For this reason, it is crucial for web business owners to learn what their customers *want*. This thesis provides a conceptual framework and an implementation of a system that helps to *better* understand the *behavior* and potential *interests* of web site visitors by accounting for both *explicit* and *implicit feedback*. This thesis is divided into two parts.

The first part is rooted in computer science and information systems and uses graph theory and an extended click-stream analysis to define a framework and a system tool that is useful for analyzing web user behavior by calculating the *interests* of the users.

The second part is rooted in behavioral economics, mathematics, and psychology and is investigating *influencing factors* on different types of web user choices. In detail, a model for the *cognitive process* of rating products on the Web is defined and an *importance hierarchy* of the influencing factors is discovered.

Both parts make use of techniques from a variety of research fields and, therefore, contribute to the area of *Web Science*.

Zusammenfassung

Einleitung

Welche Interessen verfolgen meine Webseiten-Nutzer? Diese Frage beschäftigt viele Betreiber von Online-Unternehmen. Um in einem solch hart umkämpften Markt wie dem des Internetbusiness erfolgreich bestehen zu können, ist es für die Entscheidungsträger dieser Unternehmen ausschlaggebend zu verstehen, welche Ziele ihre Kunden verfolgen. Hauptziel der vorliegenden Arbeit ist es, diese Frage mit Hilfe eines *konzeptionellen Bezugssystems* und der *Implementierung eines Systems* zu beantworten. Beide Elemente berücksichtigen sowohl das Verhalten, als auch das *explizite* und das *implizite* Feedback der Webseiten-Nutzer.

Der vorgeschlagene Lösungsansatz unterstützt Betreiber von Online-Unternehmen dabei ihre Kunden *besser zu verstehen*. Dies geschieht durch das Beobachten und Auswerten des *Kundenverhaltens*, um daraus die vermuteten *Kundeninteressen* zu berechnen. Außerdem werden, um den Prozess des Feedbackgebens besser zu verstehen, diejenigen Faktoren untersucht, die die Auswahl des Webseiten-Nutzers beim Feedbackgeben beeinflussen.

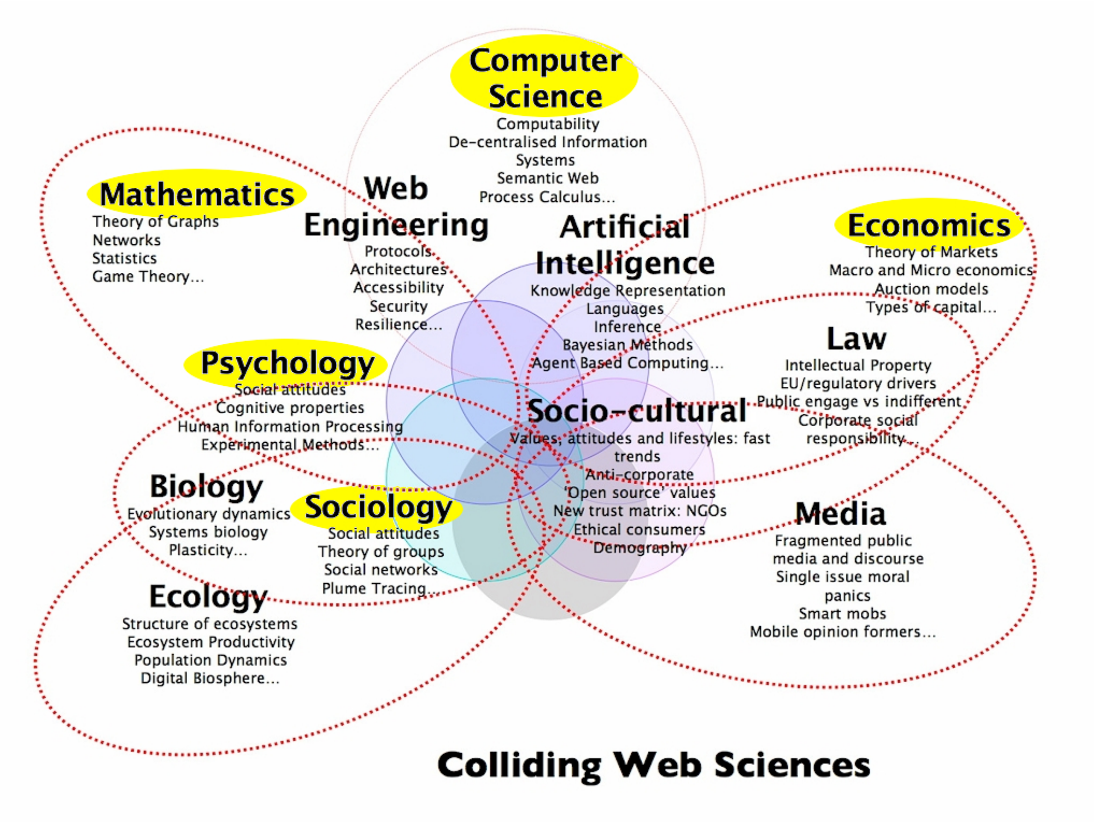
Folgende Forschungsfragen werden in dieser Arbeit im Hinblick auf unterschiedliche Aspekte des Feedbacks von Webseiten-Nutzern untersucht:

- Was lernen wir aus der Analyse des explizit und des implizit durch die Webseiten-Nutzer ausgeführten Feedbacks?

- Was sind die wichtigsten Faktoren, die das Feedback von Webseiten-Nutzern beeinflussen?

Forschungsbereich

Die Arbeit ist dem neuen interdisziplinären Bereich „*Web Science*“ zugeordnet, der kurz in diesem Kapitel vorgestellt wird. Außerdem ist die Arbeit Teil des Gugubarra-For-



Forschungsbereiche, die in dieser Arbeit zur Anwendung kommen

Abbildung 1: Die verschiedenen Forschungsgebiete der Web Science [OH08]

schungsprojektes [HKTZ06b] der Arbeitsguppe Datenbanken- und Informationssysteme (DBIS) der Goethe-Universität Frankfurt am Main. Die Arbeit steht in der Tradition früherer Veröffentlichungen um das Gugubarra-Projekt und erweitert dieses um einige neue Konzepte.

Web Science

In seinen Anfängen war die Hauptaufgabe des World Wide Web der Transport von Informationen und deren Bereitstellung. Daher war das Interesse der Wissenschaft an diesem Netz sehr eingeschränkt und auf die Bereiche der Informatik, der Mathematik und der Physik fokussiert. Die Situation änderte sich aber mit dem großen Erfolg des World Wide Web. Immer mehr Menschen begannen es zu nutzen und Web-Gemeinschaften zu bilden. Dadurch wurde das World Wide Web als Forschungsobjekt für die Sozialwissenschaften interessant. Mit dem steigenden wirtschaftlichen Erfolg des Web entstand ein

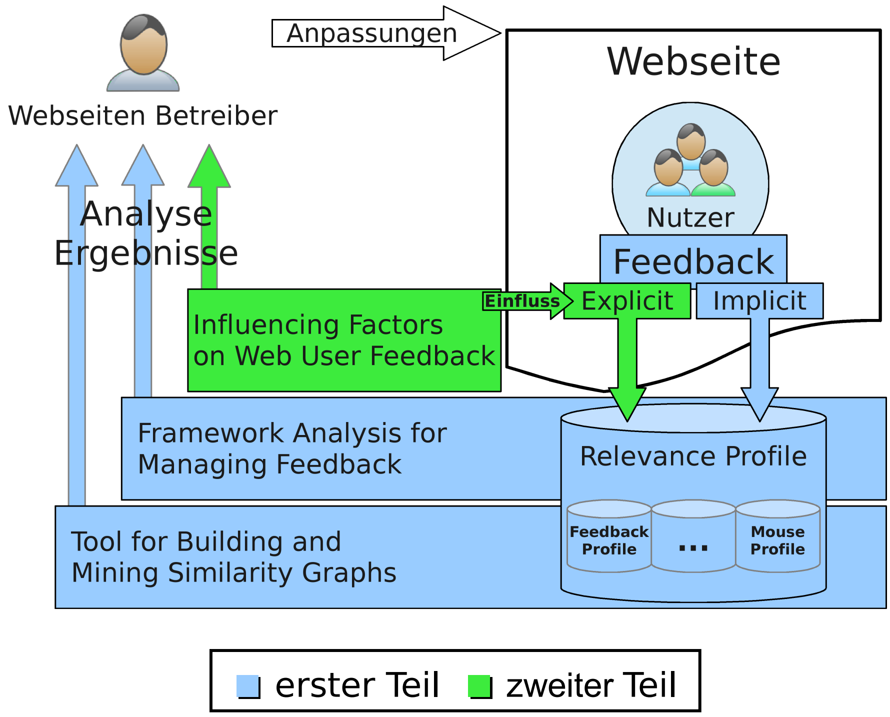


Abbildung 2: Forschungsbereich

neues und schnell wachsendes ökonomisches System, das die Aufmerksamkeit der Wirtschaftswissenschaften auf sich zog. Heutzutage entdecken immer mehr Wissenschaftler der unterschiedlichsten Disziplinen das Internet als Forschungsobjekt.

Tim Berners-Lee erkannte diesen aktuellen Trend und erschuf den Begriff der „Web Science" [BLHH+06], der dieses neue Forschungsgebiet beschreibt. Es umfasst alle Wissenschaften, die die Phänomene des World Wide Web erforschen. Dazu zählt auf der einen Seite die Forschung am „micro-level" [OH08], wie etwa die Entwicklung von mathematischen Modellen, das Entwerfen neuer Protokolle und das Entwickeln neuer Algorithmen [BLHH+06]. Auf der anderen Seite steht die Forschung am „macro-level", zu dem beispielsweise das Bloggen, Spammen oder der E-Commerce zählen. Mit den Forschungsergebnissen aus der Web Science sollen die sozialen Effekte und das Potenzial des World Wide Web vollständig verstanden werden [OH08]. Abbildung 1, entnommen aus [OH08], zeigt die verschiedenen Forschungsgebiete der Web Science und hebt gleichzeitig den interdisziplinären Charakter dieses neuen Forschungsfeldes hervor.

Da die vorliegende Arbeit Themen auch sehr unterschiedlichen Forschungsbereichen behandelt, kann sie dem Bereich der Web Science zugeordnet werden. Wie aus Abbildung 2 hervor geht, ist sie in zwei Teile gegliedert: Im ersten Teil „Inferring Web

User Feedback“ werden Methoden aus der Mathematik angewandt, um das Verhalten der Webseiten-Nutzer zu analysieren. Im Einzelnen wird mit Hilfe der mathematischen Logik die Beständigkeit des Verhaltens der Webseiten-Nutzer definiert und mit Hilfe der Graphentheorie die Bedeutung der Mitglieder der Web-Gemeinschaft berechnet. Im zweiten Teil „Influencing Factors on Web User Feedback“ kommen Methoden der Mathematik (Statistik), der Wirtschaftswissenschaften (Verhaltensöknomie), der Soziologie (Soziale Netzwerke) und der Psychologie (Kognitives Verhalten) zur Anwendung, um die Einflussfaktoren auf das Feedback in einer Web-Gemeinschaft zu analysieren. Alle Forschungsdisziplinen, die in dieser Arbeit Anwendung finden, sind in Abbildung 1 mit Gelb hervorgehoben.

Ergebnisse

Die vorliegende Arbeit stellt ein konzeptionelles Bezugssystem und die Implementierung eines Systems vor, die dabei helfen sollen, das Verhalten und die möglichen Interessen der Webseiten-Nutzer *besser* zu verstehen. Um dies zu erreichen, wird sowohl das explizite, als auch das implizite Feedback der Webseiten-Nutzer berücksichtigt. Die vorliegende Arbeit ist in zwei Abschnitte gegliedert.

Im ersten Teil - der im Forschungsgebiet der Informatik und der Informationssysteme angesiedelt ist - werden mit Hilfe der Graphentheorie und einer erweiterten Click-Stream-Analyse sowohl ein Bezugssystem, als auch ein Programm entwickelt, die aus dem Verhalten der Webseiten-Nutzer deren *Interessen* berechnen.

Im zweiten Teil - der der Verhaltensökonomie, der Mathematik und der Psychologie zugeordnet ist - werden diejenigen *Faktoren* untersucht, die unterschiedliche Entscheidungen von Webseiten-Nutzern *beeinflussen*. Im Einzelnen wird ein Modell für den *kognitiven Entscheidungsprozess* der Produktbewertung erstellt; außerdem werden die Einflussfaktoren nach ihrer jeweiligen *Priorität* eingestuft.

Mit Hilfe eines interdisziplinären Forschungsansatzes wurden die beiden folgenden Forschungsfragen untersucht:

Was können wir aus der Analyse des expliziten und des impliziten Feedbacks von Webseiten-Nutzern lernen?

Die vorliegende Arbeit trägt mit folgendem Beitrag zur Beantwortung dieser Frage bei: Um das Verhalten der Webseiten-Nutzer besser zu verstehen, wird im Kapitel 3 ein Bezugssystem (*Framework*) entwickelt, um das Feedback *einzelner* Webseiten-Nutzer zu analysieren. In sieben Schritten kann hierzu der Webseiten-Betreiber die *Ver-*

haltenskontinuität der Besucher seiner Webseite an Hand ihres Feedbacks überprüfen. Dazu wird ein neues Benutzerprofil-Konzept definiert, das sogenannte „*Relevance Profile*“ und außerdem vier *Kontinuitätsstufen*. Mit Hilfe dreier Fallstudien kann in Kapitel 5 die praktische Anwendbarkeit dieser neuen Konzepte darlegt werden. In diesen Studien werden *Verhaltensmuster* von Webseiten-Nutzern nachgewiesen, die sowohl Hinweise auf strukturelle Schwächen einer Webseite, als auch auf falsch verstandene Themen durch Webseiten-Nutzer, sein Desinteresse an einem Thema, aber auch sein anhaltendes Interesse an einem Thema geben.

In Kapitel 4 wird eine Software vorgestellt, mit der *Ähnlichkeitsgraphen erstellt und analysiert* werden können. Diese Software verschafft dem Webseiten-Betreiber einen Überblick über die Ähnlichkeiten der Interessen und die Wichtigkeit der Benutzer seine *Web-Gemeinschaft*. Die Software stellt dem Webseiten-Betreiber vier Methoden zur Auswahl, um Ähnlichkeitsgraphen seiner Web-Gemeinschaft zu erstellen. Anschließend können diese Graphen mit neun verschiedenen Algorithmen analysiert werden, von denen zwei im Zuge dieser Arbeit neu entwickelt werden. Um die praktische Anwendbarkeit dieser Software zu demonstrieren, werden in Kapitel 5 drei Fallstudien durchgeführt.

In Kapitel 6 wird die *Genauigkeit* der Benutzerprofile dadurch erhöht, dass zusätzlich zu den Server Log Dateien eine neue Quelle für die Verhaltensdaten von Webseiten-Nutzern zur Berechnung der Profile verwendet wird. Hinzu wird das Gugubarra-Framework um die Fähigkeit erweitert, die *Mausaktivitäten* der Webseiten-Nutzer zu speichern und zu analysieren (sog. „*Mouse-Tracking*“). Die Schwierigkeit besteht darin, das neue Leistungsmerkmal an die schon vorhandenen Konzepte anzupassen. Außerdem wird in einem weiteren Experiment nachgewiesen werden, dass durch die Einbeziehung der Mausaktivitätsdaten die Genauigkeit der Benutzerprofile gesteigert werden kann.

Welche sind die wichtigsten Faktoren, die das Feedback von Webseiten-Nutzern beeinflussen?

Die vorliegende Arbeit trägt mit folgendem Beitrag zur Beantwortung dieser Frage bei: In Kapitel 7 wird das Gugubarra Framework um *Leitlinien* erweitert, die eine Aussage dazu machen, wie *Bewertungssysteme* auf Webseiten *platziert und eingebettet* werden können. Dazu wird der *Einfluss*, den das *Design* verschiedener *Bewertungssystem-Skalen* auf unterschiedliche Auswahlmöglichkeiten der Nutzer hat, erforscht. Hierfür wird ein *kognitives Modell* entwickelt, um die Bewertung von Musiktiteln im World Wide Web erklären zu können. Dieses Modell besteht aus vier Komponenten und beschreibt mit Hilfe *kognitiver Heuristiken* das Auswählen und das Bewerten der Musiktitel durch

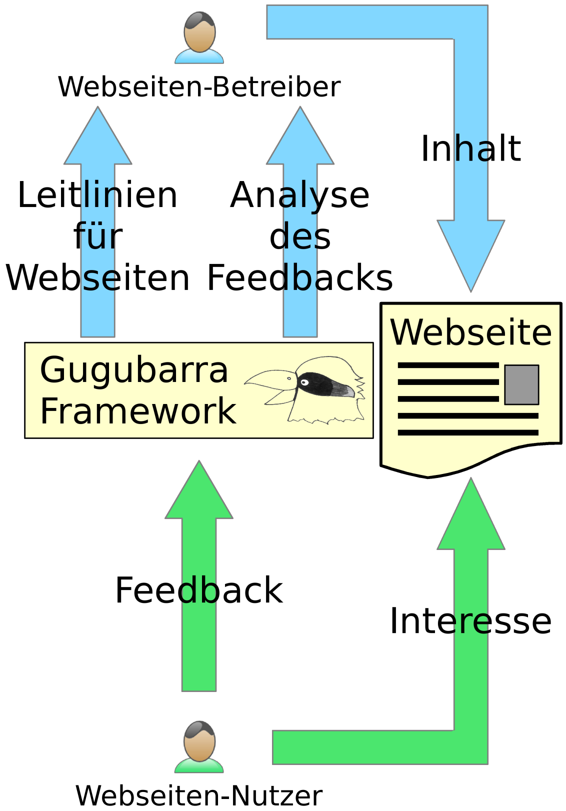


Abbildung 3: Die funktionellen Elemente des Gugubarra Framework

den Webseiten-Nutzer. Die Bedeutung der einzelnen *Einflussfaktoren* wird in drei Experimenten mit Hilfe einer Auswahl-basierten Conjoint-Analyse bestimmt. Im Einzelnen werden die Faktoren untersucht, die einen Webseiten-Nutzer beeinflussen, wenn er einen Musiktitel anhört, ihn bewertet und Interesse an ihn zeigt.

Fazit: Unterstützt durch die Ergebnisse der vorliegenden Arbeit wird ein Webseiten-Betreiber in die Lage versetzt, das Feedback sowohl *einzelner* Nutzer, als auch der *Gesamtheit* aller Mitglieder einer Web-Gemeinschaft zu untersuchen. Auch die Mausbewegungen der Webseiten-Nutzer werden hierzu analysiert. Als zusätzliche Orientierung werden neue *Leitlinien* für die Platzierung und das Einbetten von *Bewertungssystemen* auf Webseiten entwickelt. Sind dem Webseiten-Betreiber die Interessen der Webseiten-Nutzer bekannt, kann er die Inhalte der Webseiten entsprechend anpassen. Durch die Anwendung der vorgestellten Leitlinien kann er das maßgeschneiderte Feedback erfassen. Zusammengefasst zeigt Abbildung 3 alle funktionellen Elemente des Gugubarra Framework auf.

Acknowledgments

This thesis is the result of my research studies during my employment as research and teaching assistant at the Goethe-University Frankfurt am Main. I would like to thank my supervisor Professor Roberto V. Zicari for giving me the chance to work on the Gugubarra project, for the fruitful discussions in spite of his busy schedule, and for giving me the necessary time to finish this thesis. I am also grateful to Professor Miguel-Angel Sicilia for being my external reviewer.

My special thanks go to the entire Databases and Information Systems research group (DBIS) including the research staff Natascha Hoebel, Nikolaos Korfiatis, and Karsten Tolle as well as the administrative and technical staff of the group Naveed Mushtaq, Cornelia Nann, and Marion Terrell. I greatly enjoyed working with, talking to, and just being around with the DBIS team.

My appreciation goes to all students, who worked on diploma, master's and bachelor's theses that helped with the development of the Gugubarra project. Especially, I would like to thank my mouse-tracking team, Sven Eschenberg and Christian Schöneberger.

I also thank Christian Bretzinger, Manuel Giuliani, and Ines Weimer for proof-reading my thesis and for their good comments that helped me improve it.

Last but not least, I deeply thank my family for always encouraging me and supporting me during my stay in Frankfurt.

Frankfurt am Main, December 2012

Contents

Introduction

1 Introduction

1.1 Motivation

What are the interests of the visitors of my web site? This question greatly concerns many web businesses. To survive in the Internet's highly competitive environment, web companies must learn what their customers want. The main research goal of this thesis is to answer to this question by providing a *conceptual framework* and an *implementation of a system* that observes and analyzes the behavior of web site's visitors and both their *explicit* and *implicit* feedback.

The proposed approach helps web business owners to *better understand* their customers by observing and analyzing their *behavior* and by calculating their potential *interests*. Furthermore, to aid in understanding the process of web user feedback, various factors that influence the user's choices on giving feedback are investigated.

The following research questions, related to different aspects of web user feedback, are addressed in this thesis:

- What do we learn from analyzing explicit and implicit user feedback on the Web?
- What are the main factors that influence user feedback on the Web?

1.2 Research Area

The context of this thesis is the new interdisciplinary research area called "*Web Science*", which will be briefly introduced in this section. This thesis is part of the Gugubarra research project [HKTZ06b] of the Databases and Information Systems (DBIS) group at the Goethe-University Frankfurt am Main. This thesis is in the tradition of publications in the Gugubarra Framework and, consequently, extends the Gugubarra Framework with several new concepts.

1.2.1 Web Science

Historically, the World Wide Web was intended for transporting and sharing information. Therefore, scientists' interest in the Web was limited to the areas of computer science, mathematics, and physics. With the Web's success and rising popularity, the phenomenon changed. An increasing number of people began using the Internet, and social communities began to form, which attracted the social sciences to the Web. With the economic success of the Web, a fast growing, new ecosystem arose and drew the attention of the economists. Today, an increasing number of scientists from various disciplines discover the Web as an object of research.

Realizing this cross-disciplinary research interest, Tim Berners-Lee coined the term "Web Science" [BLHH+06] to describe a new field of research, which unites all sciences that discover phenomena of the World Wide Web. Beyond research on the micro-level [OH08], such as developing better mathematical models, engineering new protocols, and constructing new algorithms [BLHH+06] for the Web, the macro-level, which includes blogging, spamming, and e-commerce, must be considered to understand the social effects and potential of the Web [OH08]. Figure 1.1, taken from [OH08], shows the various areas of Web Science research, emphasizing the interdisciplinary aspects of this new field.

This thesis contributes to the area of Web Science and, therefore, covers multiple research topics from multiple research fields. The research scenario is divided into two parts, as shown in Figure 1.2. The first part, "Inferring Web User Feedback", uses techniques from mathematics to analyze the behavior of web site users. In particular, mathematical logic is used to define the consistency of users, and graph theory is used to determine the important users of a web community. The second part, "Influencing Factors on Web User Feedback", uses methods of mathematics (statistics), economics (behavioral economics), sociology (social networks), and psychology (cognitive properties) to analyze the factors that influence community feedback. The disciplines used in this thesis are marked with yellow in Figure 1.1.

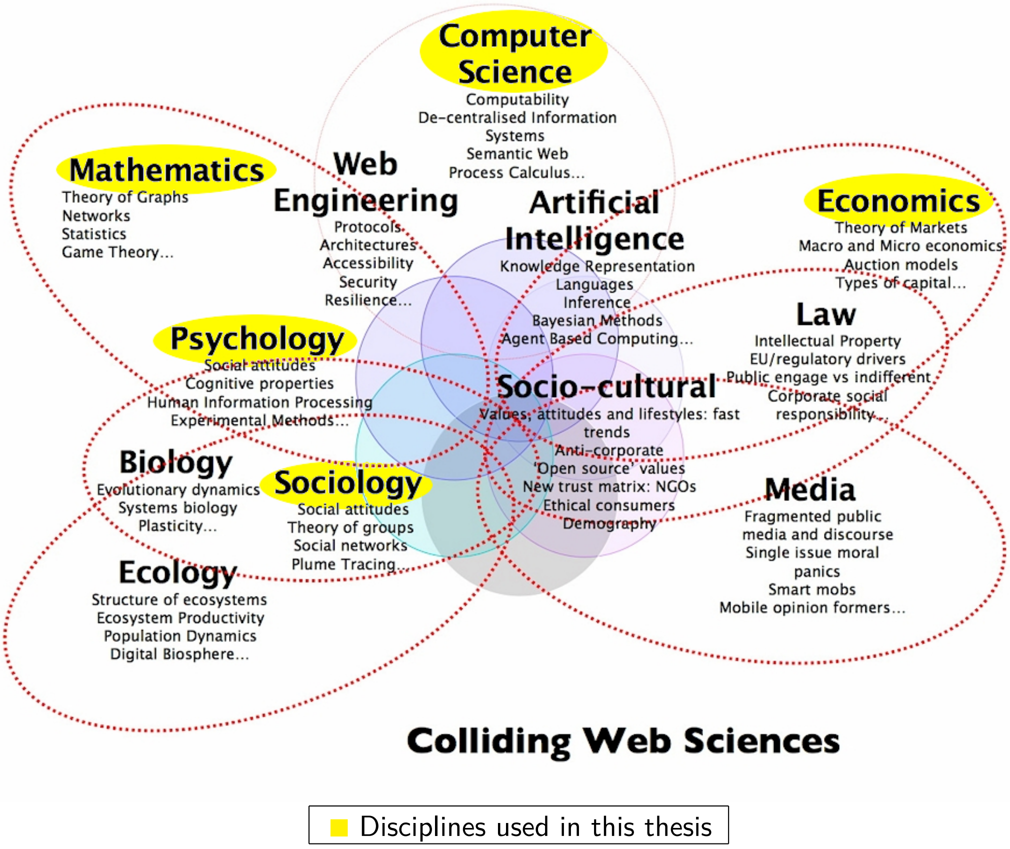


Figure 1.1: The research areas of Web Science [OH08]

1.3 Structure

The thesis is divided into two parts. The first part aims to improve the analysis functionality and accuracy of user interest profiles. The second part focuses on factors that influence web site visitors on their explicit feedback and develops guidelines for how web site business owners could place and enrich rating systems on their web site. The two parts are organized as follows:

Part I: Inferring Web User Feedback (covers the blue parts of Figure 1.2 and Figure 1.3).

Chapter 2: The Gugubarra Framework. To analyze web user feedback, we use the *Gugubarra Framework* [HKTZ06b], a web analytic tool that helps the owner of a web site to better understand the interests of the web site visitors. Chapter 2 introduces

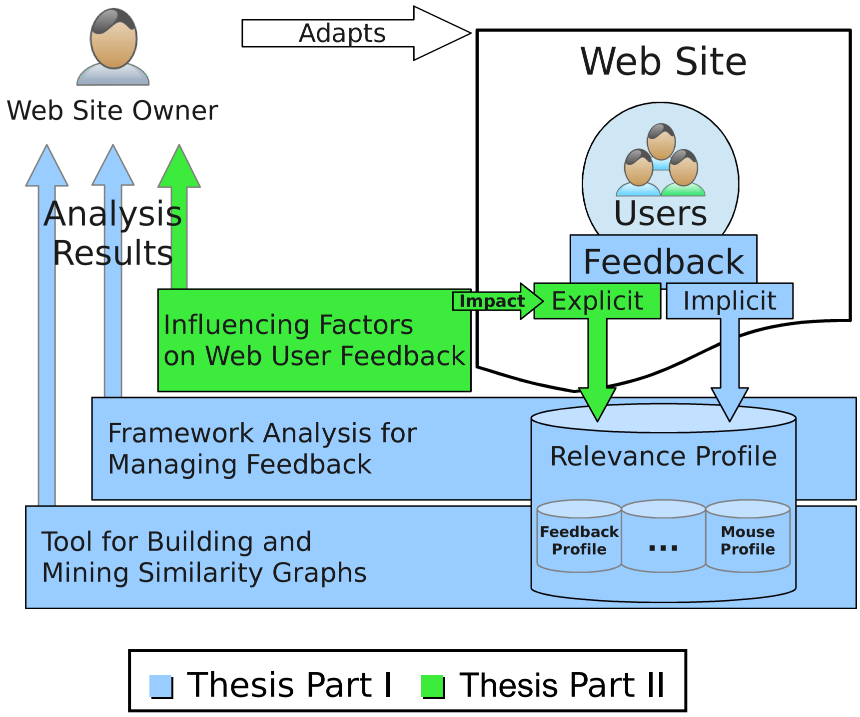


Figure 1.2: Research scenario

the Gugubarra Framework, including its *basic concepts* in Section 2.3, the different *user profiles* in Section 2.4, and the *implementation* of these concepts in Section 2.5.

Chapter 3: A Framework Analysis for Managing Feedback of Visitors of a Web Site. In Chapter 3, we use past user data to analyze the consistency of user behavior with respect to user feedback. Accordingly, we extend the Gugubarra Framework with a *framework analysis for managing the feedback of web site visitors*. Within the *seven steps* of this framework, the web site owner can identify patterns of users' interest. In detail, Section 3.6 introduces the *Relevance Profile*, and Section 3.8 introduces the methodology of the *consistency check*.

Chapter 4: User Similarity and User Importance. In Chapter 4, we analyze the on-line community in its entirety with the help of graph algorithms. In particular, we extend the Gugubarra Framework with a *tool for building and mining similarity graphs* of the web community. With this tool, the web site owner can calculate the most "important" and the most "unimportant" users with respect to their interests. This tool has a two-phase workflow. In the first phase, described in Section 4.4.1, the *similarity graph*

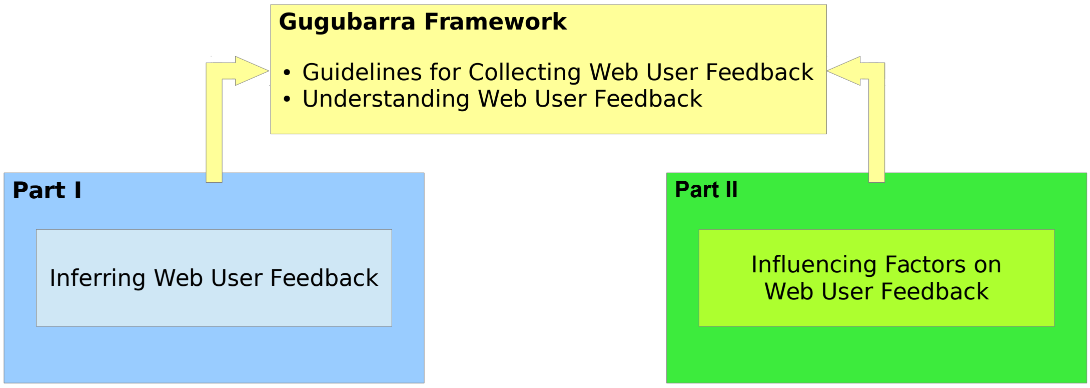


Figure 1.3: Relation of Part I and Part II

of a community, which is determined by the *similarity threshold* chosen by the web site owner, is built. In the second phase, the *importance* of the web site users is calculated. Section 4.4.2 describes nine algorithms that are used in this calculation.

Chapter 5: Case Studies. In Chapter 5, we conduct three case studies to show the *applicability* of the extended Gugubarra Framework in different real world scenarios. The chapter uses the analysis methodologies that were introduced in the prior chapters. The scenario of the first case study, described in Section 5.2, is a *cold-start situation*, which means that the user logs on the web site for the first time and the extended Gugubarra Framework calculates an initial user profile. The second case study, described in Section 5.3, compares a *cold-start situation* with a *warm-start situation*. In a warm-start situation, the user is already familiar with the web site, and her/his profile calculation includes data from previous user sessions. The third case study, Section 5.4, uses *all user data* of a real web site, collected over two years. This scenario represents the application of the extended Gugubarra Framework to a web community existing for a long period of time.

Chapter 6: Mouse-Tracking. In Chapter 6, we propose a further extension to the Gugubarra Framework, the ability to *track the mouse activities* of the web site visitors. Section 6.4 discusses the *potential* of using mouse-tracking technology in the Gugubarra Framework. The practical realization and *implementation* are presented in Section 6.5, which demonstrates the expandability of the existing concepts. In Section 6.6, we evaluate the new mouse-tracking functionality by conducting an experiment. This chapter completes the first part of the thesis, "*Inferring Web User Feedback*".

Part II: Influencing Factors on Web User Feedback (covers the green parts of Figure 1.2 and Figure 1.3).

Chapter 7: How Web Based Rating Systems Influence User's Choice. In addition to the analysis of web user behavior, in Chapter 7, we extend the Gugubarra Framework with a set of guidelines for placing and enriching rating systems [CM06] on web pages. Accordingly, we investigate the influencing factors of various rating scale designs on different types of user's choices. To measure these factors, we conduct *three experiments* on a real web store. Section 7.2 presents the theoretical insights, focusing on the *cognitive process* behind this choice. Section 7.4 presents a four-step analysis process that is based on a *choice-based conjoint (CBC) analysis* and is used in the three experiments to determine the influencing factors. The first study, described in Section 7.5, explores the impact of service attributes on the choice of *listening* to audio files. In Section 7.6, with the help of the second experiment, we determine the impact of service attributes on the choice of *rating* an audio file. The third experiment, described in Section 7.7, discovers the impact of service attributes on the choice of *indicating interest* in an audio file.

Chapter 8: Conclusion and Future Research Directions. Chapter 8 presents the conclusion in Section 8.1 and lists future research directions in Section 8.2.

How Part I and Part II relate. The goal of this thesis is to develop *a framework for web site owners* that, on the one hand, helps to understand web site visitors by *analyzing their feedback* and, on the other hand, provides guidelines for *collecting visitor feedback*. With the knowledge of the visitor's interests, the web site owner can adapt the web pages accordingly and provide proper content. With the guideline for collecting the visitors' feedback, the web site owner will be able to collect suitable feedback for the analysis. The two parts of this thesis, which are visualized in Figure 1.3, reflect the composition of this goal.

Figure 1.4 concludes the introduction by presenting a word cloud of this thesis. A word cloud is calculated from the frequencies of the words in a text without stop words, such as *is*, *as*, and *or*. Only the most frequent words are shown, and the font size of a word increases with its frequency in the source text. At the beginning of Part I and Part II, the word cloud of the respective part is shown.

Figure 1.4: Word cloud of the complete thesis

Part I

Inferring Web User Feedback

2 The Gugubarra Framework

With the Gugubarra Framework, a web based analytic tool is introduced that helps the owner of a web site to understand the interests of the web site users. In the first part of this chapter, the novel concepts introduced by Gugubarra are discussed. In the second part, the implementation of these concepts is presented.

2.1 Introduction

The web based analytic framework *Gugubarra*, also described in [MWTZ04, HZ08], is a prototype system developed by the Databases and Information Systems (DBIS) research group at the Goethe-University Frankfurt am Main. The purpose of the system is to help the owner or manager of a web site to more fully understand the interests of the registered users on her/his web site.

In this project, a *web site* is a collection of web pages, where *visitors* or *users* can register and log on. The combined group of *registered* users of this web site are called the *on-line community* or *web community*. This web site is maintained by a web site *owner* who controls the content and decides on the business strategies or goals. During a user *session*, which is defined as the time between the log-in and the log-out of a web user, all web page requests are stored in the log files of the web server and enriched with additional information, such as zones, topics, and actions, which are explained later in

Figure 2.1: Laughing Kookaburra[1]

Chapter 2. All of these data are used to calculate profiles describing the interests of every web site user.

It is assumed that a web site owner has certain business goals, e.g., to reach a certain number of reads, sales, or registered users on her/his web portal [HZ08]. For example, she/he wants to keep the users on the web site and to attract new users. To achieve these goals, the web site owner has to react to the on-line community and provide or adapt the content of the web site accordingly. Therefore, she/he has to understand the following:

- what the visitors want on the web site,
- why they visited a specific web page, and
- what the interests of the users are.

[1]This image is published by Flagstaffotos (`http://www.flagstaffotos.com.au/`) under the terms of the GFDL license (`http://www.gnu.org/copyleft/fdl.html`).

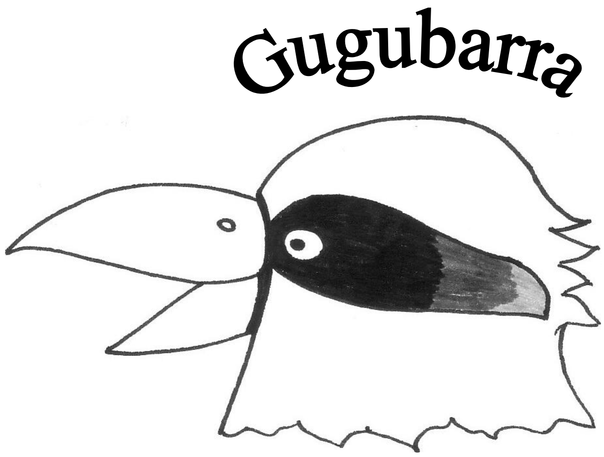


Figure 2.2: Logo of the Gugubarra project

To attain these goals, a common server log analysis, such as described by Weischeldel and Huizingh in [WH06] or by Jung et al. [JHW07], is refined using several new concepts, which will be discussed in the next sections.

The remainder of this chapter is structured as follows: In Section 2.2, privacy issues related to the Gugubarra Framework are discussed. Section 2.3 introduces the basic concepts of the Gugubarra Framework, i.e., zones, topics, and topic weights in Section 2.3.1 and actions and their weights in Section 2.3.2. Important to the basic concepts are the different user profiles of Gugubarra, which are described in Section 2.4.1, Section 2.4.2, and Section 2.4.3. The implementation of these concepts is explained in Section 2.5. This chapter ends with the conclusion and a summary of possible future work activities in Section 2.6.

Remark 2.1 *Origin of the Name "Gugubarra"*
Gugubarra is the name the Australian aboriginals use for the laughing Kookaburra bird; Figure 2.1 shows a picture of the laughing Kookaburra bird and Figure 2.2 depicts the logo of the Gugubarra project.
The name was chosen without ulterior motive, but a student (Sven Eschenberg) suggested the following meaningful explanation afterwards: The Kookaburra is a carnivorous bird that hunts for its food. The Gugubarra Framework also hunts for the data of the web site users to calculate their interest profiles.

2.2 Gugubarra and Data Privacy

With the commercialization of the Internet, the need to collect data and analyze user behavior is increasing. Many ideas in business are based on selling collected data. The standards concerning trading the data of web visitors that are in place to protect their privacy vary between countries and between companies; these standards range from very restrictive to very loose [MSB00, JK08].

With the Gugubarra Framework, both sides are accommodated. On the one hand, the web site owner who wants to earn money by using the data of the web site users often comes into conflict with the privacy protection standards. On the other hand, the users of the web site are concerned about the security of their (behavioral) data [HZ11]. The one side cannot exist without the other, i.e., without the commercial use of the behavioral data, many web sites would not be profitable. However, if a web site owner uses the data of her/his customers irresponsibly, she/he will lose them, and often, the foundation of her/his business will follow.

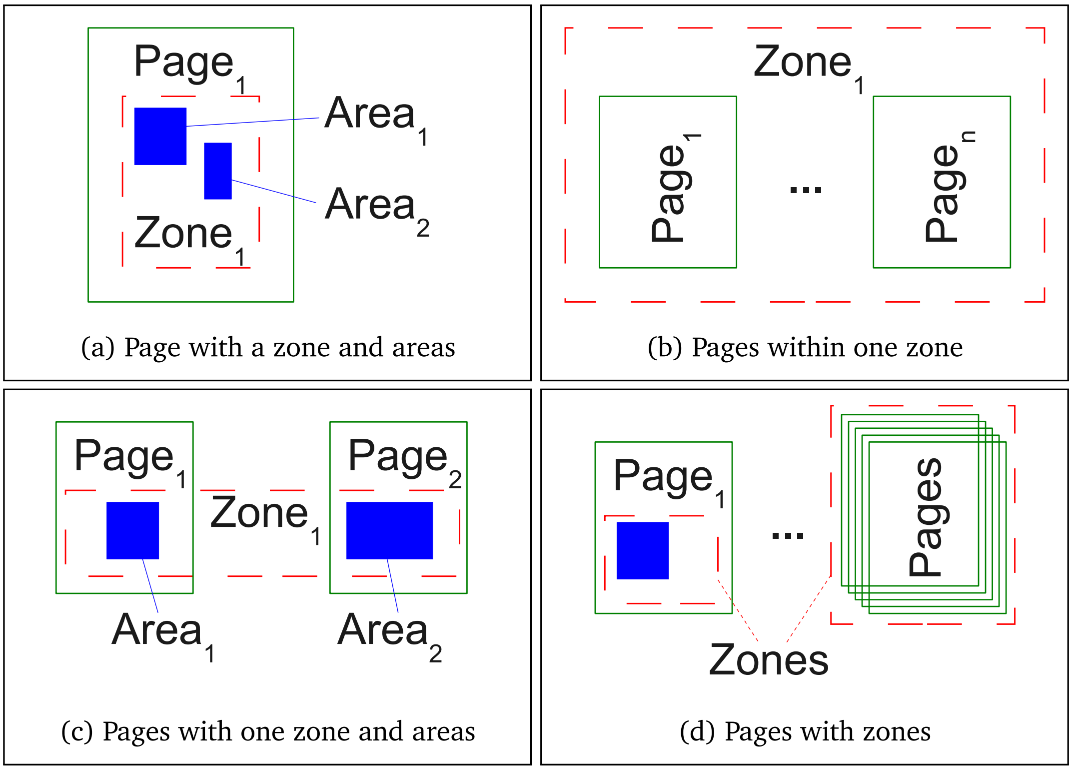
To manage the privacy issue, the Gugubarra Framework is designed for an on-line community of *registered* users. Such a finite community makes it easier for the web site owner to inform the users that their actions and behavior on the community pages are analyzed and that several user profiles are built from these data. With the awareness that they are being tracked, users can always decide not to register to this community and so avoid being tracked. However, in the end, the proper and responsible use of the data is the responsibility of the web site owner.

2.3 The Basic Concepts of Gugubarra

The concepts of Gugubarra are based on observations of bricks-and-mortar businesses, i.e., grocery stores. How these observations are transferred to the modern e-commerce world is described in the following sections.

2.3.1 Zones, Topics, and Topic Weights

Many grocery stores are organized into different *zones* [Ill07], which makes it easy for the customer to find the desired goods. Therefore, these zones contain similar items that are semantically connected. For example, there is a zone with large refrigerators that contain different types of frozen foods and another zone with shelves that are filled with a wide variety of candies. Semantically, these zones are not totally disjunctive, e.g., in the zone with frozen foods, sweet ice cream can also be found.



(a) Page with a zone and areas

(b) Pages within one zone

(c) Pages with one zone and areas

(d) Pages with zones

Figure 2.3: Gugubarra concepts: zones, areas, and pages

With the Gugubarra Framework, the concept of zones is transferred to the world of e-commerce [HKTZ06b, HKTZ06a], i.e., a *zone* of a web site consists of one (Figure 2.3a) or more web pages (Figure 2.3b) or a set of parts of a web page (Figure 2.3c and Figure 2.3d). Zones must be disjunctive and each zone has a unique ID. For simplicity, individual parts making up a zone are denoted as areas (see Figures 2.3a-2.3d). Therefore, a zone can also be defined as a set of areas.

For each zone, a list of *topics* is defined, which describes the contents of the zone. Each topic, in turn, has an associated *weight*. Topic weights are defined by the owner of the web site and are used to indicate the relative importance of the given topic with respect to the zone where the topic is defined. The topic weight is represented by a number between 0 and 1; *0* represents minimum importance, while *1* denotes maximum importance of this topic. A topic can be assigned to different zones with different weights.

In analogy to the grocery store, the zone with large refrigerators could have a "frozen food" topic with a weight of 0.9 (very important) because lot of different types of frozen foods are stored there. In the same zone, a refrigerator with ice cream can be placed and represented by the "sweets" topic with a weight of 0.2 (less important) because ice cream is only a small part of this zone. In contrast, the “sweets” topic could have a high weight in the zone with candies, e.g., 1.0, which indicates maximum importance.

2.3.2 Actions and Action Weights

Parallel to the structural observations of a grocery store, a comparison can be made to the store customers. Customers usually perform many actions during their visit, e.g., they walk around the store, take goods off the shelves, touch the goods, put the goods into their shopping cart, and pay for the goods at the cashier. Some of these actions are more important to the store owner than others. Walking could indicate a person’s interest in the range of products or only curiosity. However, a customer who is paying for goods indicates a definite interest in the store’s products.

To reflect these observations in Gugubarra, a set of *actions* that can occur within the web site are defined [HKTZ06b]. Clicking a link and downloading a file are examples of possible actions. Each action also has an associated *weight* that indicates the relative importance of the specific action in the context of the zone where the action can occur. The action weight has a value greater than or equal *0*. A value with a range between *0* and *1* has a damping effect on the weight of the topics in the zone where the action occurs. Accordingly, an action weight greater than *1* has a strengthening effect on the impact of the topics.

The next sections will show, in detail, how this information is used to calculate several user profiles of web site visitors.

2.4 User Profiles of Gugubarra

Gugubarra has different types of user profiles, i.e., user profiles for explicit feedback, user profiles for implicit feedback, and a user profile that unites all interest profiles of a user in one single profile. Figure 2.4 gives an overview of all Gugubarra user profiles, which will be explained in the next sections in more detail.

2.4.1 Explicit Feedback User Profiles

In Gugubarra the *explicit* user data are stored in *two* different profiles, the *Obvious Profile* (OP) and the *Feedback Profile* (FP) [MWTZ04]. Explicit user data means, that the web site user is directly asked by the web site owner about the data, e.g., by an e-mail or a web form. The advantages of these types of data is that they come directly from the user and that the user is aware of being asked about her/his interests. Thus, the results can

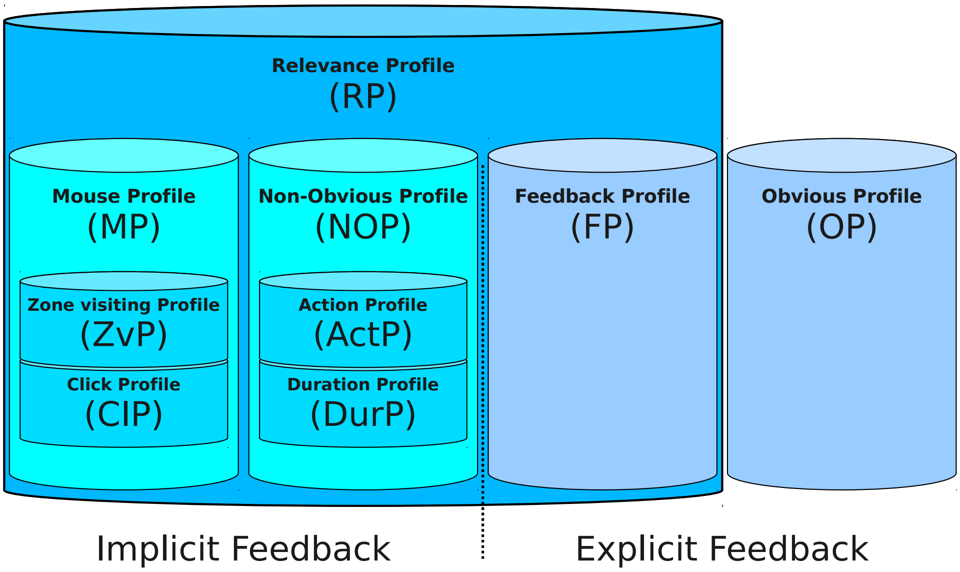


Figure 2.4: Gugubarra user profiles

reflect the interests of a user very accurately. However, the disadvantages are that a user can misinterpret the topics and/or give inaccurate answers. The explicit user feedback is a valuable source for the calculation of user interest profiles. The next sections describe the use of the explicit user feedback in the context of the Gugubarra Framework.

Obvious Profile (OP)

The *Obvious Profile* (OP) stores the data that are explicitly given by a user, e.g., name, age, address, and e-mail address, during the initial registration process, and from updates as they occur [MWTZ04]. These data are *not* used to calculate interest profiles but could be used to obtain a demographic overview of the web community.

Feedback Profile (FP)

The *Feedback Profile* (FP) stores the data that are explicitly given by a user about her/his interests in the topics of the web site [MWTZ04].

Technically, all FPs are vectors and store the *interests* of each user, u_m, related to a topic T_i at time t_n. Each row contains the calculated interest values of the user for a topic. The values of the calculated interests are between 0 and 1, where 1 indicates high interest, and 0 indicates no interest in a topic or that the user did not give any explicit feedback about her/his interest in this topic. Figure 2.5 displays an example of a FP of a user, showing a high interest in topic T_2 with an interest value of (1.0), lower interest in topic T_1 (0.3), and no interest in topic T_3, (0.0).

$$FP_{u_m,t_n} = \begin{pmatrix} 0.3 \\ 1.0 \\ 0.0 \end{pmatrix} \begin{array}{l} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{array}$$

Figure 2.5: FP for a user u_m defined for three topics T_1, T_2, and T_3

2.4.2 Implicit Feedback User Profiles

In addition to the explicit user data, the Gugubarra Framework calculates user interests from the implicit user data. The sources of the implicit user data are the interactions of the visitors with the web site, particularly, the behavioral data. With these data, Gugubarra compensates for the constraints of the explicit user data mentioned in Section 2.4.1. The implicit user data are stored in the *Non-Obvious Profile* (NOP), which consists of the *Action Profile* (ActP) and the *Duration Profile* (DurP) [HKTZ06b]. In Chapter 6, the implicit user profiles of the Gugubarra Framework are extended with data from the mouse activities of the web site user.

Action Profile (ActP)

The *Action Profile* (ActP) takes into account the activities a user performs in a given zone with respect to a topic and is defined as follows:

$$ActP(T_i) = \frac{\sum_q (\sum_t aw_t * v(T_i, Z_q))}{\sum_s aw_s} \tag{2.1}$$

For each zone Z_q with topic T_i, the associated topic weight, v, is multiplied with the sum of the weights of all actions, aw_t, that occur in that zone. The result is then normalized by the sum of the weights of all actions that have occurred. The results are normalized to obtain values between 0 (no interest) and 1 (high interest) for the specific topic T_i.

Similar to the FP (see Figure 2.5), the ActP stores the calculated interests of a user in a vector. The stored values are between 0 and 1, where 1 indicates high interest, and 0 indicates no interest in a topic or that the user did not perform any action in a zone with this topic.

Duration Profile (DurP)

The *Duration Profile* (DurP) takes into account, for a topic T_i, the *time (duration)* spent by the user on a page that contains that topic. It is calculated as follows:

$$DurP(T_i) = \frac{\sum_j (duration(P_j) * v(T_i, P_j))}{\sum_k duration(P_k)} \tag{2.2}$$

For a topic T_i, the weights v of the topic T_i are summed for all pages P, for the zones that contain the topic T_i. The result is then multiplied by the time the user has spent on each page ($duration$). Finally, the result is normalized by dividing it by the total time the user spent on the web site.

Similar to the FP (see Figure 2.5), the DurP stores the calculated interests of each user in a vector. The stored values are between 0 and 1, where 1 indicates high interest, and 0 indicates no interest in a topic, or that the user did not spend any time (duration) on a page with this topic.

Non-Obvious Profile (NOP)

The *Non-Obvious Profile* (NOP) combines the ActP and DurP to calculate the behavioral data of a user of the web site.

The calculation of a NOP for a user, u_m, at a given time, t_n, with respect to a topic, T_i, is automatically computed by Gugubarra as follows:

$$NOP_{u_m,t_n}(T_i) = a * ActP(T_i) + b * DurP(T_i) \tag{2.3}$$

In the calculation of the NOP, the importance of the *actions* (expressed by the ActP) and the *time duration* (expressed by the DurP) can be determined, relative to one another, by setting the two parameters a and b in the formula. We can assign values between 0 and 1 to a and b, with the condition that their sum must be 1. For example, with a setting of the two parameters $a = 0.8$ and $b = 0.2$, the ActP would have a much higher impact than the DurP on the NOP calculation.

Similar to the FP (see Figure 2.5), the calculated interests of a user are stored in a vector. The value of each element is between 0 and 1, where 1 indicates high interest, and 0 indicates no interest in a topic, or that no data are available about the user's interest in that specific topic.

Mouse Profile (MP)

In Chapter 6, the Gugubarra Framework will be extended with the capability to track the mouse interactions of a user with the web site. Therefore, the *Mouse Profile* (MP) will

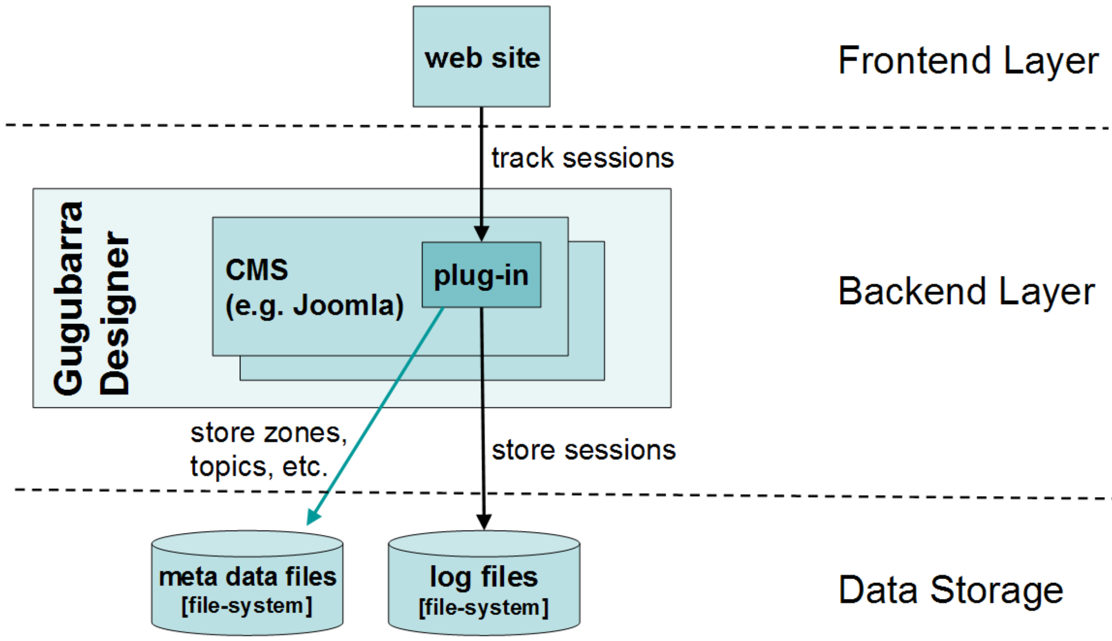


Figure 2.6: Architecture of the Gugubarra Designer [Hoe11]

be introduced. It consists of the *Zone visiting Profile* (ZvP) and the *Click Profile* (ClP). Further details will be discussed in Chapter 6.

2.4.3 The Relevance Profile (RP)

The *Relevance Profile* (RP) unites the explicit and the implicit feedback profiles of a user into a single interest profile. This profile will be defined and discussed in Chapter 3 of this thesis. With the help of the RP, the web site owner can calculate the consistency of the users and also collect valuable information about the community.

2.5 Implementation

In this section, the implementation of the two parts of the Gugubarra Framework, the Gugubarra Designer and the Gugubarra Analyzer, will be briefly discussed. A more detailed description can be found in the PhD thesis of Natascha Hoebel [Hoe11]. Below, version 3.0 of the Gugubarra Framework is presented.

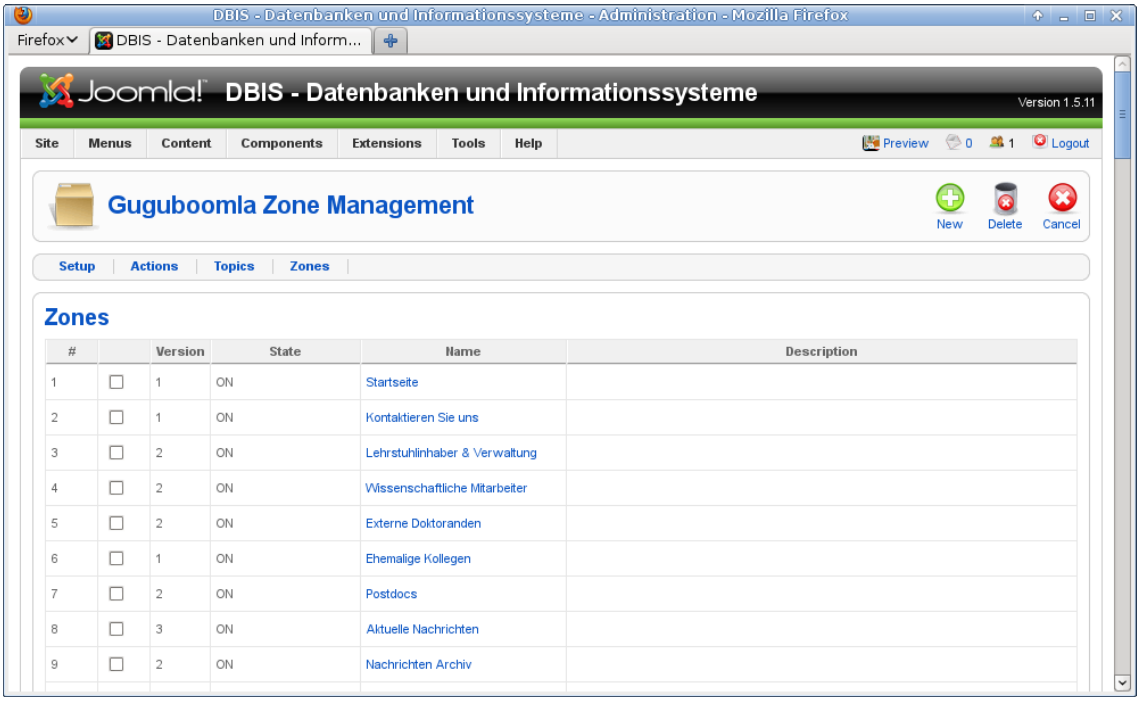


Figure 2.7: The Gugubarra Designer Joomla! plug-in showing the zone management

2.5.1 Gugubarra Designer

The *Gugubarra Designer* helps the web site owner to include the concepts of Gugubarra into the web site. Figure 2.6 shows the layer view of the architecture of the Gugubarra Designer. The frontend layer is made up of the web site that is analyzed with the Gugubarra Framework.

Currently, web sites are usually composed of more than one web page, and are often very complex. The use of a content management system allows the web site owner to administrate the web site more easily. Therefore, the Gugubarra Designer is built as a plug-in for a content management system. This plug-in forms the backend layer of the Gugubarra Designer. In [Hoe11], the full description of the construction of the plug-in for the Joomla![2] content management system, Guguboomla, can be found. Guguboomla is the reference implementation for the Gugubarra Designer used in this thesis. With this plug-in, the web site owner can add Gugubarra zones, topics, and actions, with their corresponding weights, to the web site. Figure 2.7 presents a screenshot of the Gugubarra Designer with its zone management section.

All these Gugubarra meta data are stored in files on the web server. The meta data

[2] `https://www.joomla.org/`

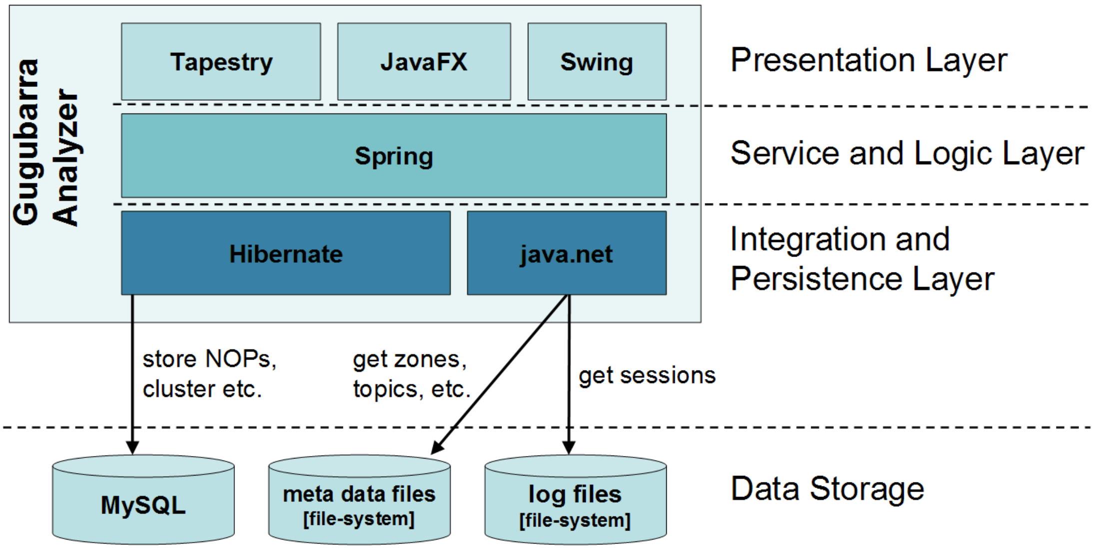


Figure 2.8: Architecture of the Gugubarra Analyzer [Hoe11]

files are coded using the Extensible Markup Language (XML)[3] document format that are good readable by humans too. The user sessions are managed by the content management system, i.e., Joomla!, and stored on the same server as the server log files.

In this thesis, the Gugubarra Designer is extended with the ability to track the mouse activities of the web site visitors; see Chapter 6. An advanced feedback form system for the collection of the explicit user feedback is also added; see the bachelor thesis of Solaimankhel [Sol10].

After the integration of the Gugubarra concepts into the web site, the behavioral data of registered web site visitors are collected and analyzed by the Gugubarra Analyzer, which is introduced in the next section.

2.5.2 Gugubarra Analyzer

The main tasks of the *Gugubarra Analyzer* are to analyze the data of the Gugubarra Designer, to build the user profiles, and to provide the web site owner with a web application to analyze her/his web community. Figure 2.8 shows the layers of the architecture of the Gugubarra Analyzer with frameworks used for the implementation. The Gugubarra Analyzer is a separate service and can be installed on the same or a different machine than the Gugubarra Designer.

[3]http://www.w3.org/XML/

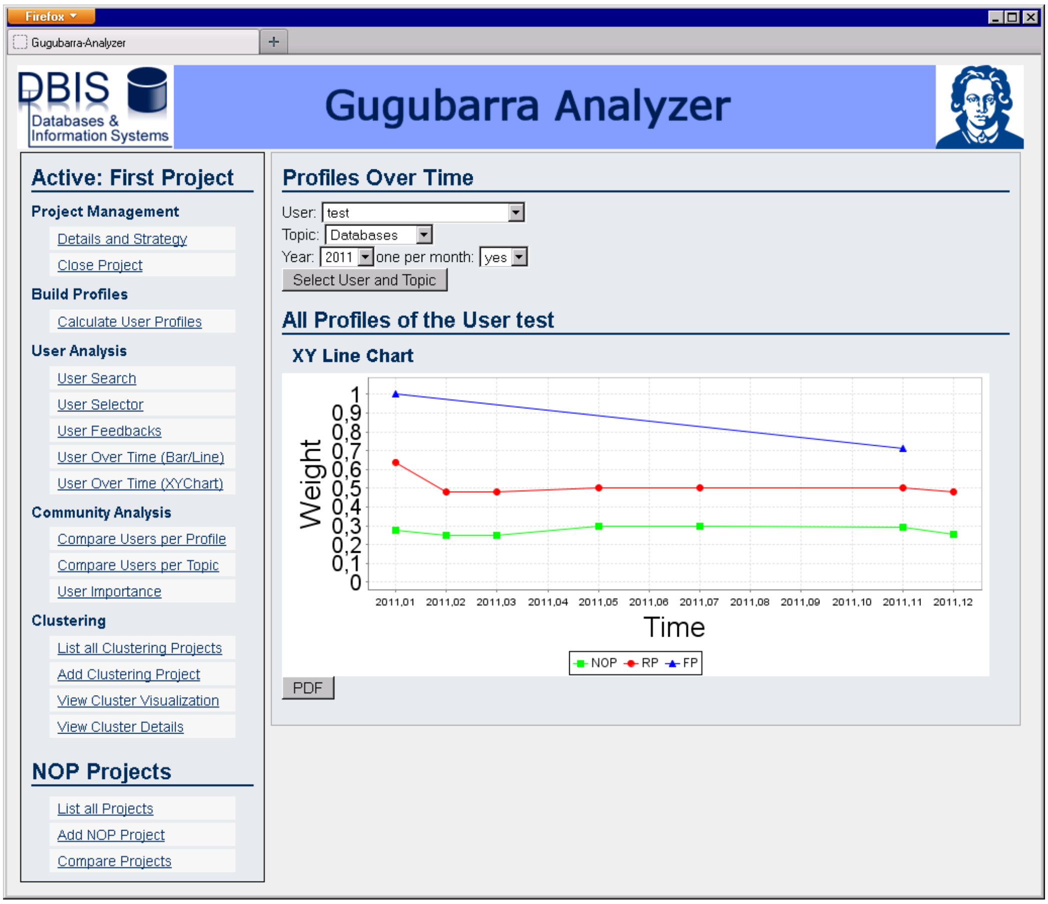


Figure 2.9: The Gugubarra Analyzer web application

The GUI[4] of the web application is the presentation layer. Tapestry[5], an open source framework for creating dynamic web applications in Java[6], is used in this application to build the simple HTML[7] pages, such as the configuration dialogs. For more complex pages, the Java libraries Swing and JavaFX are used. Within the GUI configuration dialogs, the web site owner can influence the user profile calculations by changing different parameters. After configuration, the different user profiles are calculated and presented to the web site owner.

The Gugubarra Analyzer comes with a few of different analysis services, which allow the web site owner to examine some statistics about the web community. For example, Figure 2.9 presents the NOP, the FP, and the RP of a user for one topic and for one year.

[4]Graphical User Interface
[5]`https://tapestry.apache.org/`
[6]`http://www.java.com/`
[7]HyperText Markup Language, `http://www.w3.org/html/`

The user profiles, the graph calculation, and the different services are part of the service and logic layer. All these services are provided using Spring[8], a Java framework for the development of enterprise applications.

The next layer, the integration and persistence layer, provides the business objects, i.e., zones, topics, and users, to the Gugubarra Framework. The needed data are queried from the database and mapped to objects using Hibernate[9], a framework for the storage and retrieval of Java domain objects via object/relational mapping. This layer also organizes the access and download of any meta data or log files on Gugubarra Designer that are located on another server. The access and download occur via HTTP[10] and are implemented with the Java library Java.net.

The bottom layer is the data storage layer. In this layer, the downloaded meta data and log files are stored on the Gugubarra Analyzer server, parsed, and written into a MySQL[11] relational database for further use.

In this thesis, the Relevance Profile, described in Chapter 3, and the Mouse Profile, described in Chapter 6, calculations are also integrated into the Gugubarra Analyzer. The user profiles analysis tools are extended so that the web site owner is supported by conducting a consistency check, as presented in Chapter 3. A tool is implemented to build and mine similarity graphs of her/his on-line community, as discussed in Chapter 4. With this tool, the web site owner can determine the most important and unimportant users of the community using different algorithms.

2.6 Conclusion and Future Work

With the Gugubarra Framework, a tool is developed to support web site owners in understanding their communities. In this chapter, the Gugubarra Framework, its basic concepts, and its implementation, are introduced. With the Gugubarra concepts of zones, topics, and actions, new ideas are implemented to refine the common click stream analysis. Furthermore, the web site owner can fine tune these concepts and so influence the user profiles calculations to reflect the needs of her/his business strategy by defining the weights of the topics and the actions. The Gugubarra Framework has three different types of user profiles. The first type is built from the explicit user feedback, and the second type is calculated from the implicit user feedback. The last type is a container profile, the RP, which is calculated from all other user interest profiles.

[8] `http://www.springsource.org/`
[9] `http://www.hibernate.org/`
[10] Hypertext Transfer Protocol
[11] `http://www.mysql.com/`

After the introduction of the basic concepts, the implementation of the Gugubarra Framework is presented. The Gugubarra Framework consists of two parts. The first is the Gugubarra Designer, built as a plug-in for a content management system that allows the web site owner to integrate the Gugubarra concepts into the web site. The other part, the Gugubarra Analyzer, provides several services to analyze the on-line community.

In future work, especially with respect to user feedback, the Gugubarra Framework could be enhanced by Web 2.0 technologies. The web site visitors could become involved in the process of the topic assignment to zones, i.e., with tags, users would be able to add new topics to zones. Tags are meaningful keywords, added by users to a web resource, such as an image. These words can describe the content of the resource, in this case, the zone. The result is a flat taxonomy [GH06] for the tagged resource, which is built by the users (the folk). Thus, tags are sometimes called folksonomy [VW07]. The weight of user topics could be determined by simply counting how often the same tag word was added to a zone. This approach would move Gugubarra from a web owner-centric perspective to a relatively web user-focused perspective. However, this approach would also imply a loss of power of the web site owner over her/his web site.

Table 2.1: Contribution summary

Subject	***Detail***
Gugubarra Designer	We included the ability to track the mouse activities of the users into the web analytics system Gugubarra (Section 2.5.1)
	We extended the explicit feedback system with an advanced feedback form system (Section 2.5.1)
Gugubarra Analyzer	We implemented the Relevance Profile (RP) calculation (Section 2.5.2)
	We integrated the Relevance Profile into the analysis services of the Gugubarra web application (Section 2.5.2)
	We implemented the Mouse Profile (MP) calculation, including the Click Profile (ClP) calculation and the Zone visiting Profile (ZvP) calculation, in the Gugubarra Framework (Section 2.5.2)
	We extended the web application to support the web site owner in performing a consistency check (Section 2.5.2)
	We extended the web application with more analysis services (Section 2.5.2)

3

A Framework Analysis for Managing Feedback of Visitors of a Web Site

In this chapter, a framework analysis for managing the feedback of web site visitors is presented. Within the seven steps of this framework, the web site owner can identify patterns of users' interest. Moreover, the Relevance Profile and the methodology of the consistency check are introduced. This chapter is based on the publications of Schefels and Zicari [SZ10, SZ12].

3.1 Introduction

An important issue in the management of a web-based user community, where users are registered to a web portal, is to identify *patterns of users' interest*. In this context, the users' feedback plays a major role.

In this chapter, we present a framework analysis for managing the feedback given by registered visitors of a web site. We use as a reference system Gugubarra (introduced in Chapter 2), a prototype system developed by DBIS at the Goethe-University of Frankfurt, for analyzing the interests of web site users.

The rest of the chapter is structured as follows: Section 3.2 recalls the Gugubarra Feedback Profile and introduces the seven steps of the framework for managing user feedback. From Section 3.3 to Section 3.9 these seven steps of the framework are discussed in detail. Section 3.10 presents related work and Section 3.11 outlines the conclusion and future work.

In this chapter, we consider the problem of how to manage user feedback. We look at the notion of *consistency*, that is, to what extent the behavior of a user (expressed by her/his NOP) differs from her/his declaration of interest given with an explicit feedback. We assume that a user is willing to give a feedback, indicating values associated to a list of topics. This is obviously not the only way for a user to give an explicit feedback, but we will use this method for simplicity in the rest of this chapter. We are aware of the limitations of explicit user feedback as indicated in [JF03], where studies of eBay's reputation system have shown that it is difficult to elicit user feedbacks without some sort of incentive.

In the literature (see Section 3.10, Related Work), a user feedback is often used in the field of information retrieval for ranking search results to predict user preferences. This chapter, in contrast, uses feedback in the context of integrating it in the user profile, which is independent from a search.

In the rest of the chapter, we assume that a user is aware and has given permission that a interest profile is generated automatically and is kept for her/him.

3.2 User Feedback Profile

We assume, we ask a user for an explicit feedback (see step 3 below), by asking the user to define her/his interest with respect to a set of predefined topics, giving a numerical value between 0 and 1 for each topic: with 0 indicating no interest, and 1 indicating much interest for a specific topic T_i. To capture this user information, we define a *Feedback Profile* (FP) as a vector, similar in structure to a NOP. Figure 3.1 shows an example of such FP, for the same user u_m, and for the same three topics, T_1, T_2, and T_3.

$$FP_{u_m,t_n} = \begin{pmatrix} 1.0 \\ 0.5 \\ 0.0 \end{pmatrix} \begin{array}{l} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{array}$$

Figure 3.1: FP for a user u_m, defined for three topics T_1, T_2, and T_3

We can see from the example, that the user has given as a feedback different interest values for T_1, T_2, while confirming the no interest in T_3. We assume in the rest of the chapter, that we capture the explicit feedback given by a user in her/his associated FP.

Our framework for managing user feedback for a community of registered users of a web site is composed of several steps:

- Step 1: Definition of a scope
- Step 2: Definition of a filter
- Step 3: Obtaining explicit user feedback
- Step 4: Filtering the user feedback
- Step 5: Clustering
- Step 6: Consistency check
- Step 7: Interpreting the results of the consistency check

Each step is detailed in the rest of this chapter.

3.3 Step 1: Definition of a Scope

The meaning of this initial step is to analyze for which users and for which topics (*scope*) we want to take into consideration the users' feedback. With this step, we define a *scope* for the users' feedback, that is, we set a *cluster of users*, and a *cluster of topics* for which we apply the *filtering* (step 2) of the user's feedback.

A cluster of users can be defined in different ways: for example based on their behavioral patterns, and/or based on the user's personal data, etc. No matter how the cluster of users is defined, we impose the condition that a cluster of users has at least one user in it. In the extreme case it may contain all registered users.

A cluster of topics can also be defined in a different way: e.g., giving a priority list to the list of topics defined for the web site, and/or taking into account a possible hierarchy or classification of topics, etc. Here, we assume for simplicity that a cluster of topics is a subset of the list of all topics defined for a web site. Such a cluster may contain no topics.

3.4 Step 2: Definition of a Filter

In order to define the *relative importance* of a user feedback, we introduce in this step the notion of a *filter*, as a function f_l, which results in a value between 0 and 1. If the result value of f_l is near to 1, then the user feedback has a substantial influence for the associated cluster of users and/or topics. While a low value of f_l, close to 0, decreases the influence of the user feedback. With $f_l = 0$, the user feedback is *not* taken into account for the associated cluster of users and/or topics. With f_l, we decide here how much impact the user feedback should have for a given cluster of users and/or topics.

f_l is a function defined for a *scope*, which is a given cluster of users and a given cluster of topics, resulting in a value between 0 and 1:

$$f_l(\underbrace{Cluster_{u_m},\ Cluster_{T_i}}_{scope\ S_k}) \rightarrow between\ 0\ and\ 1 \tag{3.1}$$

We can apply the same function f_l with different resulting values to different input scope, i.e., $Cluster_{u_m}$ and $Cluster_{T_i}$. This is a flexible way to handle user's feedback.

For example, if the $Cluster_{u_m}$ contains all registered users, then the same filter is set for all users, that is, we treat the feedback in the same way for all users. Nevertheless, if the $Cluster_{u_m}$ contains a *subset* of all registered users, this means we apply the specific filter value only for the users who belong to the input cluster. If we want to treat the feedback of users belonging to another cluster in a different way, we simply apply the filter with a different return value to them.

Moreover, if we want to handle in a special way the user feedback related to a *specific* set of topics, we simply group the topics in a cluster and apply a specific filter value. In this way, we can, for example, decide that the user feedback related to a specific topic T_1 is more important (or less important), than the one given for another topic T_2.

3.5 Step 3: Obtaining Explicit User Feedback

The process of obtaining an explicit user feedback may vary. In this chapter, for simplicity we restrict our attention to three cases:

- The user is presented a set of topics $T_1 \ldots T_n$, and it is asked to give feedback by indicating a number between 0 (no interest) and 1 (very interested) for each topic in the list. The user feedback is then stored as it is, in a Feedback Profile, FP, associated to the user.

- The user is first shown the value of her/his automatically calculated Non-Obvious Profile for a set of topics $T_1 \ldots T_n$, and then it is asked to give a feedback indicating a number between 0 (no interest) and 1 (very interested) for each topic in the NOP. The user feedback is then stored as it is, in a separate Feedback Profile, FP, associated to the user.

- The user is presented with her/his last Feedback Profile (FP) for a set of topics $T_1 \ldots T_n$, and it is asked to give a feedback, indicating a number between 0 (no interest) and 1 (very interested) to each topic in the FP. The user's feedback is then stored as it is, in the Feedback Profile, FP, associated to the user.

It is not in the scope of this chapter to discuss the different semantics and implications of these different ways of soliciting users' feedback. We plan to study this research issue in a forthcoming study.

3.6 Step 4: Filtering the User Feedback

We consider the scope and the filter set for the user feedback, and we create for each user who has given a feedback and for which we have a NOP, a new profile called *Relevance Profile* (RP). A Relevance Profile is a vector containing values between 0 and 1. The process of applying the scope and filter to the user's feedback works as follows:

At a given time t_n, given a scope S_k defined by a cluster of users $Cluster_{u_m}$, and a cluster of topics $Cluster_{T_i}$, and a filter f_l for it. For each user u_m, belonging to the scope S_k, and given a set of topics T_i, also belonging to the scope S_k, we create a *Relevance Profile*, related to the topics T_i, using the following formula:

$$RP_{u_m,t_n}(T_i) = \frac{NOP_{u_m,t_n}(T_i) + f_l(S_k) * FP_{u_m,t_n}(T_i)}{a + b + f_l(S_k)} \tag{3.2}$$

The RP is calculated by integrating the two available interest profiles of the user: the Non-Obvious Profile (NOP), which is automatically calculated, based upon the behavior of the user (i.e., her/his Action Profile and her/his Duration Profile), with the explicit feedback given by the user indicated in her/his Feedback Profile (FP), and filtered f_l to the relevant scope S_k.

In the formula, RP sums up the NOP applying the filter to FP, and it is normalized to obtain values between 0 and 1, which corresponds to the domain of the NOP.

After this step, we have for each registered users who has given a feedback, besides a NOP and FP also an RP. The benefits of RP is that it integrates in a flexible way into one single user profile both the implicit feedback (calculated from the behavior data) and explicit feedback of the user. In Section 3.11, we will show a few areas of applicability for the RPs.

For a very high number of users, it is better to work with clusters of RP rather than with RPs of individual users, as it is illustrated in the next section.

3.7 Step 5: Clustering

With this step, we apply a clustering algorithm to the generated user profiles. This is necessary when the number of registered users is very high (e.g., millions of users) and the analysis of individual user interest is no more possible. In Gugubarra, we implemented a *fuzzy clustering algorithm* [HK10, KLL04] to create *clusters of Relevance Profiles*. RPs can be clustered together in several different ways: for example grouping together users with RPs showing similar behavioral patterns of interest. We recall that an RP integrates the behavioral data calculated by the NOP, and the filtering of the user feedback stored in the FP.

The results of clustering users RPs depends on the value we have set for the *filter*. If all RPs are generated with a filter $f_l = 0$, we did not take into consideration the explicit user feedback (if any) in the calculation of the RPs. If we use a filter $f_l = 0$, a *cluster* represents RPs taking into account only the calculated behavioral data (implicit user feedback) stored in the NOP of the users, and not the explicit user feedback (if any). If we want taking into account the explicit user feedback, we have re-calculate the RPs by setting a filter different from 0 and cluster again. Furthermore, it is possible to cluster only Feedback Profiles (FPs).

Our approach is quite flexible, as we generate RPs by scope, using different filter f_l values.

3.8 Step 6: Consistency Check

In *step 6*, we turn our attention to the notion of user *consistency*. That is, we look if the behavior of a user (calculated and expressed by her/his NOP) is *consistent* with her/his declaration of interest, expressed in her/his explicit feedback (captured in the FP), and filtered and generated in her/his RP.

We do not intend to judge the intention of the users but rather measure the possible *discrepancy* between actions performed by a user (implicit feedback) and her/his declaration of interest expressed by the explicit feedback. We will show in Section 3.11 some possible domain of applicability of our approach for measuring user *consistency*.

In Gugubarra, the interest of a user in a topic is encoded as a number between 0 and 1, which can be represented by a metric scale. The interpretation of such a scale is not very intuitive. To measure the consistency of a user or a cluster of users, we divide the metric scale into n intervals. These intervals have not to be equally distributed but should be disjunctive. At least, we name each interval corresponding to an ordinal scale. The intervals should cover the whole metric interest scale.

Definition 3.1 *D is an ordered set of n ordinal categories cat_x:*

$$D = \{cat_1, cat_2, \ldots, cat_n\}$$
$$with\ cat_1 \neq cat_2 \neq \ldots \neq cat_n$$

Definition 3.2 *Each category cat_x is assigned to an interval on a metric interest scale:*

$$cat_x = [a, b] = \{y \in \mathbb{R} \mid a \leq y \leq b\}$$
$$with\ a, b \in \mathbb{R}\ and\ 0 \leq a, b \leq 1\ and\ cat_1 \cap cat_2 \cap \ldots \cap cat_N = \emptyset$$

The number n of the ordinal categories defines the granularity of the ordinal scale. The granularity level will influence the results of the consistency check. An example for such a classification would be:

$$\begin{aligned} D = \{&\text{“}low\ interest'', \text{“}medium\ interest'', \text{“}high\ interest''\} \text{ with} \\ &\text{“}low\ interest'' = [0.0, 0.3), \\ &\text{“}medium\ interest'' = [0.3, 0.7), \text{ and} \\ &\text{“}high\ interest'' = [0.7, 1.0] \end{aligned}$$

In order to have a measurable indication of user consistency, we compare for a given user, her/his calculated RP with respect to her/his NOP and FP of a certain topic. We define the term *consistency* as follows:

We define three propositional logic variables [Sch95] called R, F, and N for a given user u_m in addition to her/his three interest profiles *NOP*, *FP*, and *RP*. These variables describe the “*location*” of the interest profiles of a user in respect to the ordinal scale as defined:

Definition 3.3 *Location:*
A user's interest profile for a topic T_i is located *into category cat_x, if its value is a (proper) subset of the interval of cat_x, denoted with e.g., $RP_{u_m,t_n}(T_i) \subseteq cat_x$.*

Definition 3.4 *Location Variables:*

$$\begin{aligned}
&R = RP_{u_m,t_n}(T_i) \subseteq cat_x \\
&F = FP_{u_m,t_n}(T_i) \subseteq cat_x \\
&N = NOP_{u_m,t_n}(T_i) \subseteq cat_x \\
&with\ cat_x \in D
\end{aligned}$$

Furthermore, four definitions of consistencies are given, taking into account R, F, and N:

Definition 3.5 *Explicit Consistent (C^e):*
When the RP *and the* FP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge F$) but not in the same category with the* NOP *($\neg N$), we call this user or this cluster of users* explicit consistent.

$$R \wedge F \wedge \neg N \rightarrow C^e$$

Definition 3.6 *Implicit Consistent (C^i):*
When the RP *and the* NOP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge N$) but not in the same category with the* FP *($\neg F$), we call this user or this cluster of users* implicit consistent.

$$R \wedge \neg F \wedge N \rightarrow C^i$$

Definition 3.7 *Total Consistent (C^t):*
When the RP, *the* FP, *and the* NOP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge F \wedge N$), we call this user or this cluster of users* total consistent.

$$R \wedge F \wedge N \rightarrow C^t$$

Definition 3.8 *Total Inconsistent ($\not{C}^t$):*
When the RP*, the* FP*, and the* NOP *of a given user u_m or a cluster of users at time t_n are each located into different ordinal categories ($\neg$ R $\wedge$ $\neg$ F $\wedge$ $\neg$ N), we call this user or this cluster of users* total inconsistent.

$$\neg R \wedge \neg F \wedge \neg N \rightarrow \not{C}^t$$
$$or \quad \neg C^t \rightarrow \not{C}^t$$

Definitions 3.5 and 3.6 say that a user is explicit or implicit consistent if her/his FP *or* NOP is in the same ordinal category like her/his RP. Accordingly, one interest profile is always in another (i.e., in a higher or a lower) ordinal category. For example, if a user is explicit consistent, her/his NOP is in a different ordinal category than the FP and the RP (see Definition 3.6). This extra piece of information can also be included into our definitions of explicit and implicit consistency. Therefore, we add to the three location variables, defined in (3.4), two new location variables $F^{\uparrow}$ and $N^{\uparrow}$ for a given user u_m:

Definition 3.9 *Location Variables $F^{\uparrow}$, $N^{\uparrow}$ with cat_x, $cat_y \in D$:*

$$F^{\uparrow} = FP_{u_m,t_n}(T_i) \subseteq cat_y \text{ with } cat_x \leq cat_y$$
$$N^{\uparrow} = NOP_{u_m,t_n}(T_i) \subseteq cat_y \text{ with } cat_x \leq cat_y$$

The superscript arrow indicates that a profile is in a higher ordinal category, e.g., $F^{\uparrow}$ means that the FP of a user is in a higher category than the other profiles. Therefore, the negated form (e.g., $\neg$ $F^{\uparrow}$) means the contrary: the profile is in a lower ordinal category compared with the other interest profiles of a user. Now, we can extend the definitions of the explicit and implicit consistencies:

Definition 3.10 *Up Explicit Consistent ($C^{e\uparrow}$):*
When the RP *and the* FP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x (R $\wedge$ F) and the* NOP *is in a higher category and, therefore, in a different category ($\neg$ N $\wedge$ $N^{\uparrow}$), we call this user or this cluster of users* up explicit consistent.

$$R \wedge F \wedge \neg N \wedge N^{\uparrow} \rightarrow C^{e\uparrow}$$
$$or \quad C^e \wedge N^{\uparrow} \rightarrow C^{e\uparrow}$$

Definition 3.11 *Down Explicit Consistent ($C^{e\downarrow}$):*
When the RP *and the* FP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge F$) and the* NOP *is in a lower category and, therefore, in a different category ($\neg N \wedge \neg N^{\uparrow}$), we call this user or this cluster of users* down explicit consistent.

$$R \wedge F \wedge \neg N \wedge \neg N^{\uparrow} \rightarrow C^{e\downarrow}$$
$$\text{or} \quad C^{e} \wedge \neg N^{\uparrow} \rightarrow C^{e\downarrow}$$

Definition 3.12 *Up Implicit Consistent ($C^{i\uparrow}$):*
When the RP *and the* NOP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge N$) and the* FP *is in a higher category and, therefore, in a different category ($\neg F \wedge F^{\uparrow}$), we call this user or this cluster of users* up implicit consistent.

$$R \wedge \neg F \wedge N \wedge F^{\uparrow} \rightarrow C^{i\uparrow}$$
$$\text{or} \quad C^{i} \wedge F^{\uparrow} \rightarrow C^{i\uparrow}$$

Definition 3.13 *Down Implicit Consistent ($C^{i\downarrow}$):*
When the RP *and the* NOP *of a given user u_m or a cluster of users at time t_n are located into the same ordinal category cat_x ($R \wedge N$) and the* FP *is in a lower category and, therefore, in a different category ($\neg F \wedge \neg F^{\uparrow}$), we call this user or this cluster of users* down implicit consistent.

$$R \wedge \neg F \wedge N \wedge \neg F^{\uparrow} \rightarrow C^{i\downarrow}$$
$$\text{or} \quad C^{i} \wedge \neg F^{\uparrow} \rightarrow C^{i\downarrow}$$

3.9 Step 7: Interpreting the Results

The factual information about the consistency of behavior of users, computed at *step 6*, is valuable information that can be used to better understand the user community and the *effectiveness* of a web site. With *step 7*, we attempt to give an interpretation of the results from the consistency check. A general interpretation is difficult because there are many factors to be considered: beginning with the composition of the on-line community of the web site, the content and layout of the web site, and the tuning of the parameters of the Gugubarra Framework.

3.10 Related Work

User feedback is often used in the field of information retrieval, for ranking search results to predict user preferences. White et al. [WJR01] examine the extent to which implicit feedback can act as a substitute for explicit feedback, where searchers explicitly mark documents as relevant. With our approach, we combine NOP and FP in one profile and use this data for analysis.

Agichtein et al. [ABD06] examine alternatives for incorporating feedback into the ranking process and explore the contributions of user feedback compared to other common web search features. The implicit feedback causes significant improvements on the quality of web search result rankings. With the NOP we integrate implicit feedback into the RP to improve its expressiveness.

Lin et al. [LLYT05] define a so called *reputation* manager, which collects feedback ratings from its clients after each transaction. In our approach, web site owners can easily weight the feedback of users with respect to their reputation. For example, using a different filter f_l for expert and non-expert users or with a specific consistency history, we can get different weights reflecting their reputation levels.

In [PB97], Pazzani and Billsus define algorithms for learning and revising user profiles that can determine, which World Wide Web sites on a given topic would be interesting to a user. The authors use a Bayesian classifier for this task.

In [JGP+05], Joachims et al. examine the reliability of implicit feedback generated from click through data in web search. They analyze the users' decision process using eye tracking and compare the implicit feedback against manual relevance judgments, and conclude that clicks are informative but biased.

Zigoris and Zhang [ZZ06] use a Bayesian adaptive user profiling with explicit and implicit feedback, and address the cold-start problem, proposing that implicit feedback should be combined with explicit feedback to get a stable base for prediction. This is related to our future work where we want to use the RP for predicting user interest changes.

Fink, Kobsa, and Schreck [FKS97] define a public accessible, personalized hypermedia system with an adaptive user interface and content. They also distinguish between obvious and non-obvious user profile information. In addition, to overcome the cold-start problem, they use so called “stereotypes”.

With the advent of Web 2.0, large social communities like Facebook[1] or Last.fm[2] became popular. These communities with their high number of users related data, often use distributed databases and frameworks to store and analyze these data. Shmueli-Scheuer et al. present in [SSRC+10] a method to extract user profiles from large scale data. This method is based on the Apache Hadoop[3] MapReduce framework. In contrast to our work, they only use implicit user feedback extracted from the web server logs to calculate the user profile. However, their work focuses on the scalability of their method and could be interesting for our future work since our target groups are also large web communities.

3.11 Conclusion and Future Work

The main result of this chapter is the definition of a framework analysis for managing user feedback. This framework integrates user behavioral data, automatically computed, and explicit user feedback, filtered into the Relevance Profile (RP). The calculation of RPs, their interpretation, and applicability was presented with the help of examples.

Applicability of Relevance Profiles. The flexible way by which we manage the user feedback by incorporating it, filtered together with the NOP data, into a single coherent profile per user (the Relevance Profile) makes it attractive for a variety of practical applications. Here, we list some of them:

RPs are useful to solve the so called “*cold-start problem*” defined by Maltz and Ehrlich in [ME95]: if a new user registers to the web site, at the beginning we have no information about her/his interests since the user did not perform any action yet. With the integration of the user Feedback Profile into her/his RP, we can solve this problem by asking the user about her/his interests. If the user is willing to give a feedback at the time of the registration, we integrate it into her/his RP.

RPs are well suited to *cluster and compare* users, with no need to consider several different profiles for the same user.

RPs make it easy to add new forms of user feedbacks (e.g., *mouse-tracking* Chapter 6, *eye-tracking* [YJSH08]) by integrating them into the RPs, and setting a filter defined reusing different available analysis methods.

[1]https://www.facebook.com/
[2]http://www.last.fm/
[3]http://hadoop.apache.org/

RPs are useful in cases of *missing information*. Sometimes there is missing information about a topic in the user profile, for example, because the user never visited a page with this specific topic. When the web site owner asks the user for an explicit feedback and she/he indicates a high interest in this topic, it is possible that the user did not find the topic on the web site. A possible reason could be that the web site has a bad design or bad navigation, so that the user was not able to find the available web pages with this topic of interest. If the explicit feedback indicates instead a low interest in these topics, the user is simply not interested and did, therefore, not visit the web pages containing this topic. Which means, the missing interest information is not caused by a bad designed web site. In conclusion, by integrating the explicit feedback into her/his RP allows the web site owner to evaluate how effective the web site is.

Explicit feedback error compensation is another domain where RPs are useful. For example, a user could interpret an ambiguous topic term in a feedback questionnaire different than the web site owner intended. This could result in that the user indicates high interest in a topic she/he is not interested, misled by the ambiguous topic term. The RP can compensate this kind of errors by taking into account the NOP, the behavioral data of the user. In detail, if a user is not interest in a topic, she/he will not visit many web page with this topic. Therefore, the RP corrects the interest value of the user in a misunderstood topic and compute a lower interest value for this topic (how much lower depends on the setting of the filter function f_l, see Section 3.3 and Section 3.4).

In future, we want to analyze different factors influencing the feedback of a user, e.g., which type of questionnaire forms [VSL07] should be used (see Chapter 7), how often should we ask for feedback, and what new Web 2.0/Web 3.0 techniques, for example tagging [Jaz07], can support feedback collection. Moreover, we want to use the RPs as a basis for *predicting changes in users' interests*. To gain reasonable prediction results, we want to include more user interest profiles into the RP. We are currently implementing a mouse-tracking module for our Gugubarra Framework (see Chapter 6). With these new data source we will have an extra source for the prediction of users' interests.

Table 3.1: Contribution summary

Subject	***Detail***
Framework for Managing Feedback	Step 1: Definition of a scope, i.e., select the topics and the users for the RP calculation (Section 3.3) Step 2: Definition of a Filter, i.e., weighting the explicit feedback of the users in a flexible way (Section 3.4) Step 3: Obtaining Explicit User Feedback (Section 3.5) Step 4: Filtering the explicit user feedback by setting the scope and the filter function (Section 3.6) Step 5: Clustering the user profiles for the interpretation (Section 3.7) Step 6: Consistency Check, i.e., comparison of the different feedback profiles of a user to detect discrepancies between the users action and explicit feedback (Section 3.8) Step 7: Interpreting the results (Section 3.9)
Relevance Profile (*RP*)	– Unites implicit and explicit feedback of a user into one interest profile (Formula 3.2) – Explicit feedback can be filtered by the scope and the filter function f_l
Interest Category	A set of ordinal categories projected to intervals on a metric interest scale (Definition 3.2)
Consistency	Explicit Consistent (C^e) (Definition 3.5) Implicit Consistent (C^i) (Definition 3.6) Total Consistent (C^t) (Definition 3.7) Total Inconsistent ($\not{C}^t$) (Definition 3.8)

4
User Similarity and User Importance

In this chapter, the Gugubarra Framework is extended with a tool for building and mining similarity graphs. With this tool, the most important and the most unimportant users of an on-line community can be calculated. Most parts of this chapter are based on the publications of Schefels [Sch12, Sch13].

4.1 Introduction

Nowadays, web-based user communities enjoy great popularity. The social network Facebook[1] has more than 1 billion active users [fac13] and even the relatively new Google+[2] about 235 million [Gun12]. In this highly competitive environment, it is crucial for web site owners to understand and satisfy their web community.

Previous research discovered community structures in these networks but focused only on the pure friendship structure of these communities [EBB10]. In this chapter, we present a tool for building and mining similarity graphs. These similarity graphs are built from the interest profiles of the users of a web community. We use the Gugubarra Framework [MWTZ04, HZ08], introduced in Chapter 2, to build interest profiles of web users.

[1] `https://www.facebook.com/`
[2] `https://plus.google.com/`

As described in Section 2.3, in Gugubarra, each user profile is stored as a vector that presents the supposed interests of a user u_m related to a topic T_i at time t_n. Each vector row contains the calculated interest value of the user for a given topic. The values of the interest are between 0 and 1, while 1 indicates high interest and 0 indicates no interest for a topic (see Example 4.1).

Example 4.1 *Relevance Profile RP of user u_m for three topic T_1, T_2, T_3:*

$$RP_{u_m,t_n} = \begin{pmatrix} 0.3 \\ 1.0 \\ 0.0 \end{pmatrix} \begin{array}{l} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{array}$$

To measure the similarity of the users, we are using different techniques from graph theory. First, we will introduce the similarity threshold that helps the web site owner in building the similarity graph of her/his community. This threshold sets how similar the users must be to be connected together in the similarity graph. In addition to that, it reduces the complexity of the graph [Pal12]. Second, we will provide several algorithms to find important users in the similarity graph. There exists not only one valid definition for importance of users because it depends–as always–on the point of view. For this reason, we provide nine algorithms to discover the importance of users. Two of these algorithms are new designed in respect to the needs of similarity graphs.

The rest of the Chapter is structured as follows: Section 4.2 introduces basic concepts and definitions that will be used in the rest of the chapter. In Section 4.3, the similarity of users is defined. Section 4.4 presents the main contribution of this chapter, the analysis tool for building and mining similarity graphs. Section 4.5 presents the conclusion and future work of this chapter.

4.2 Basic Concepts and Definitions

In this section, we introduce the definitions of the user equality and the user similarity, concepts of the graph theory, and seven algorithms to determine the importance of users of an on-line community.

4.2.1 Similarity measurement

Due to the fact that the RP contains all information about the interests of the users, we want to use it to compute the similarity between the interests of *all* users. First, we have to define the equality of users:

Definition 4.1 *Two users u_i and u_j are equal in respect to a topic T_r of a web site at time t_n if the interest values of T_r of their RPs are equal:*

$$RP_{u_i,t_n}(T_r) = RP_{u_j,t_n}(T_r) \ where \ i \neq j.$$

To compare users, we need a measurement for *similarity*. Similarity measurements are very common in the research field of data mining. For example, documents are often represented as feature vectors [YL03], which contain the most significant characteristics like the frequency of important keywords or topics. To compute the similarity of documents, the feature vectors are compared with the help of distance measurements: the smaller the distance of the documents is the more similar they are.

Gugubarra interest profiles, i.e., the RP, can be considered as feature vectors of the users, too. They contain the most significant characteristics of the users, e.g., the interests in different topics of a web site. Therefore, we can use the similarity measurements of data mining theory to compute similarity between the members of the on-line community.

An important requirement on the similarity measurement algorithm is its performance because an on-line community can cover lots of users. Consequently, we have to choose a similarity measurement with a high performance so that the analysis program will scale with the high number of users. Aggarwal et al. proved in [AHK01] that the *Manhattan Distance*, also known as *City Block Distance* or *Taxicab Geometry*, is very well suited for high dimensional data. We share in [HMS+09] that web sites may have up to 100 topics. Thus, we have to deal with very high dimensional feature vectors, i.e., one dimension per topic (see Example 4.1).

The Manhattan Distance (L_1-norm) [Cha07] is defined as follows:

Definition 4.2 *The Manhattan Distance d (L_1-norm):*

$$d_{Manhattan}(a,b) = \sum_i |a_i - b_i|$$

with $a = RP_{u_m,t_n}$, $b = RP_{u_r,t_n}$ and $m \neq r$.

4.2.2 Graph Theory

In this section, we present the basic definitions of graph theory, which was founded by Leonhard Euler [Eul36], that are necessary for our tasks.

A *graph* G [Wil79] is a tuple $(V(G),\ E(G))$. $V(G)$ is a set of *vertices* of the graph and $E(G)$ is the set of *edges*, which connects the vertices[3]. Two connected vertices are called *adjacent* and *incident* to the connected edge. Accordingly, an edge is a pair of two vertices v, w with $v,\ w \in V(G)$. If the graph has an empty set of vertices $V(G)$ it is called *null graph* and is often denoted by Ω [EF70]. The number of out- and incoming edges of a vertex are called *degree*, e.g., in Figure 4.1a the vertex with number 139 has a degree of eight and the vertex 115 has a degree of one.

A graph G can be represented [Bol98] by an *adjacency matrix* $A = A(G) = (a_{ij})$. This $n \times n$ matrix, n is the sum of the vertices of G, is defined as follows:

Definition 4.3 *Adjacency Matrix A of G:*

$$a_{ij} = \begin{cases} 1 & \textit{if } \{v, w\} \in E(G) \\ 0 & \textit{otherwise.} \end{cases}$$

with $v,\ w \in V(G)$

In a *simple graph*, an edge connects always *two* vertices [RN10]. This means that $E(G)$ consists of unordered pairs $\{v, w\}$ with $v, w \in V(G)$ and $v \neq w$ [Wil79]. In a social network, vertices could represent the members of this network and the edges could stand for the friendship relation between these vertices–so friends are connected together.

Every pair of distinct vertices of a *complete graph* [Wil79] are connected together.

The connections between edges can be *directed* or *undirected*. In a directed graph, the edges are an ordered pair of vertices v, w and can only be traversed in the direction of its connection. This means that a *simple graph* is undirected. This feature is very useful, e.g., to model news feed subscriptions of a user in a social network, a one-way friendship.

A *loop* is a connection from a vertex to itself [Bol98]. A loop is not an edge.

Labeled vertices make graphs more comprehensible. Vertices can be labeled with identifiers, e.g., in the social network graph with the names of the users.

[3] Sometimes it is postulated [Wil79] that $V(G)$ and $E(G)$ has to be finite but there exists also definitions about infinite graphs [Jun94]. However, the number of web site users should be finite.

In the same way edges can be labeled to denote the type of connection. In the social network graph example, the label could represent the type of relation between users, e.g., friend or relative.

With *weighted graphs*, the strength of the connection between the single vertices can be modeled. Every edge has an assigned weight. In a social network, the weight could be used to display the degree or importance of the relationship of the users. A weighted graph can also be represented by an adjacency matrix (see Definition 4.2 above) where a_{ij} is the weight of the connection of $\{v, w\}$. See Example 4.2 for an adjacency matrix of a similarity graph of five users:

Example 4.2 *Adjacency matrix A:*

$$A = \begin{pmatrix} 0.00 & 1.28 & 1.19 & 2.79 & 1.18 \\ 1.28 & 0.00 & 1.63 & 2.83 & 1.90 \\ 1.19 & 1.63 & 0.00 & 2.50 & 1.35 \\ 2.79 & 2.83 & 2.50 & 0.00 & 2.85 \\ 1.18 & 1.90 & 1.35 & 2.85 & 0.00 \end{pmatrix}$$

In this adjacency matrix, every number represents the weight of the edges between two vertices, e.g., $a_{2,4} = 2.83$ represents the edge weight of the two vertices with the numbers 2 and 4. The diagonal of this matrix is 0.00 because the graph has no loops. In an undirected graph the adjacency matrix is symmetric.

A vertex w is a *neighbor* of vertex v if both are connected via the same edge. The neighborhood of v consists of *all* neighbors of v. In a social network a direct friend is a neighbor and all direct friends are the neighborhood.

A *path* [Sch07] through a graph G is a sequence of edges $\in E(G)$ from a starting vertex $v \in V(G)$ to an end vertex $w \in V(G)$. If there exists a path form vertex v to w both vertices are connected. The number of edges on this path is called *length* of the path and the *distance* between v and w is the length of the shortest path between these two vertices. A path with the same start and end point is called *cycle*. Two vertices v and w are *reachable* from each other if there exists a path with the start point v and the end point w. If all vertices are reachable from every vertex the graph is called *connected*.

G' is a *subgraph* [Bol98] of G if $V(G') \subset V(G)$ and $E(G') \subset E(G)$. G is then the *supergraph* of G' with $G' \subset G$.

A *community* in a graph is a *cluster* of vertices. The vertices of a community are dense connected.

4.2.3 Importance

There exist many algorithms to measure the *importance* of a vertex in graph. We introduce seven of the most common algorithms:

Sergin Brin and Lawrence Page [BP98] use their *PageRank* algorithm to rank web pages with the link graph of their search engine Google[4] by importance. This algorithm is scalable on big data sets (i.e., search engine indices). Usually, the PageRank algorithm is for unweighted graphs but there exists also implementations for weighted graphs [NA08]. Pujol et al. [PSD02] developed an algorithm to calculate the reputation of users in a social network. The results of the comparison of their algorithm with the PageRank show that the PageRank is also well suited for reputation calculation, i.e., importance calculation.

The *Jaccard similarity coefficient* [Jac12] of two vertices is the number of common neighbors divided by the number of vertices that are neighbors of at least one of the two vertices being considered [Csa10]. Here, the pairwise similarity of all vertices is calculated.

The *Dice similarity coefficient* [Csa10] of two vertices is twice the number of common neighbors divided by the sum of the degrees of the vertices. Here, the pairwise similarity of all vertices is calculated.

Nearest neighbors degree calculates the nearest neighbor degree for all vertices of a graph. In [BBPSV04], Barrat et al. define a nearest neighbor degree algorithm for weighted graphs.

Closeness centrality [Fre78] measures how many steps are required to access every other vertex from a given vertex.

Hub score [Kle99] is defined [Csa10] as the eigenvector of AA^T where A is the adjacencies matrix and A^T the transposed adjacencies matrix of the graph.

Eigenvector centrality [Bon87, Csa10] corresponds to the values of the first eigenvector of the adjacency matrix. Vertices with high eigenvector centralities are those, which are connected to many other vertices which are, in turn, connected to many others.

In Chapter 5, we present evaluations of these algorithms and compare the results with two new algorithms.

4.3 User Similarity

In Gugubarra, the RP provides the most significant information about a user, which is calculated from all implicit and explicit feedback profiles. To calculate *user similarity*,

[4] https://www.google.com/

we take the RP interest value of every topic of each user and calculate the Manhattan Distance between all users of the web community as shown in the following example:

Example 4.3 *Let us assume we have a web site with three topics T_1, T_2, and T_3. This web site has two registered users u_1 and u_2. The RPs of the two users were calculated at time t_1:*

$$RP_{u_1,t_1} = \begin{pmatrix} 1.0 \\ 0.5 \\ 0.0 \end{pmatrix}, RP_{u_2,t_1} = \begin{pmatrix} 0.6 \\ 0.8 \\ 0.2 \end{pmatrix}$$

The Manhattan Distance is calculated as follows:

$$d_{Manhattan}(RP_{u_1,t_1}, RP_{u_2,t_1}) =$$

$$= |1.0 - 0.6| + |0.5 - 0.8| + |0.0 - 0.2| = 0.9$$

where 0.9 *is the distance of the interests of the both users, i.e., the similarity.*

In general, the *smaller* the calculated distance is the *more similar* are the compared users to each other. Our focus is on a large group of users (i.e., the whole web community) and not only on a single user or on a single topic. The following sections should clarify research questions such as:

- Which users are important for the on-line community?
- Which users have similar interests?
- How similar are the interests of the users of the on-line community?
- How is this specific community structured?

By answering these questions, we want to give the web site owner a useful tool to enhance her/his marketing strategies, in respect of the work of Domingos and Richardson [DR01], and rise as consequence the click rates of her/his portal.

4.4 Analysis of Similarity Graphs

We developed a new tool for building and analyzing similarity graphs. We integrated several algorithms from different research areas for the analysis of the graphs. This tool is written in R[5]. R is an open source project with a huge developer community. The archetype of R is the statistic programming language S[6] and the functional programming language Scheme[7]. R has a variety of libraries with many different functions for statistical analytics. For graph analysis, R provides two common libraries: the Rgraphviz[8] and the igraph[9] library. We are using the latter for our implementation because it provides more graph analytics algorithms[10] [Mar11] and it is better applicable for large graphs. The igraph library is available for other programming languages (e.g., C, Python).

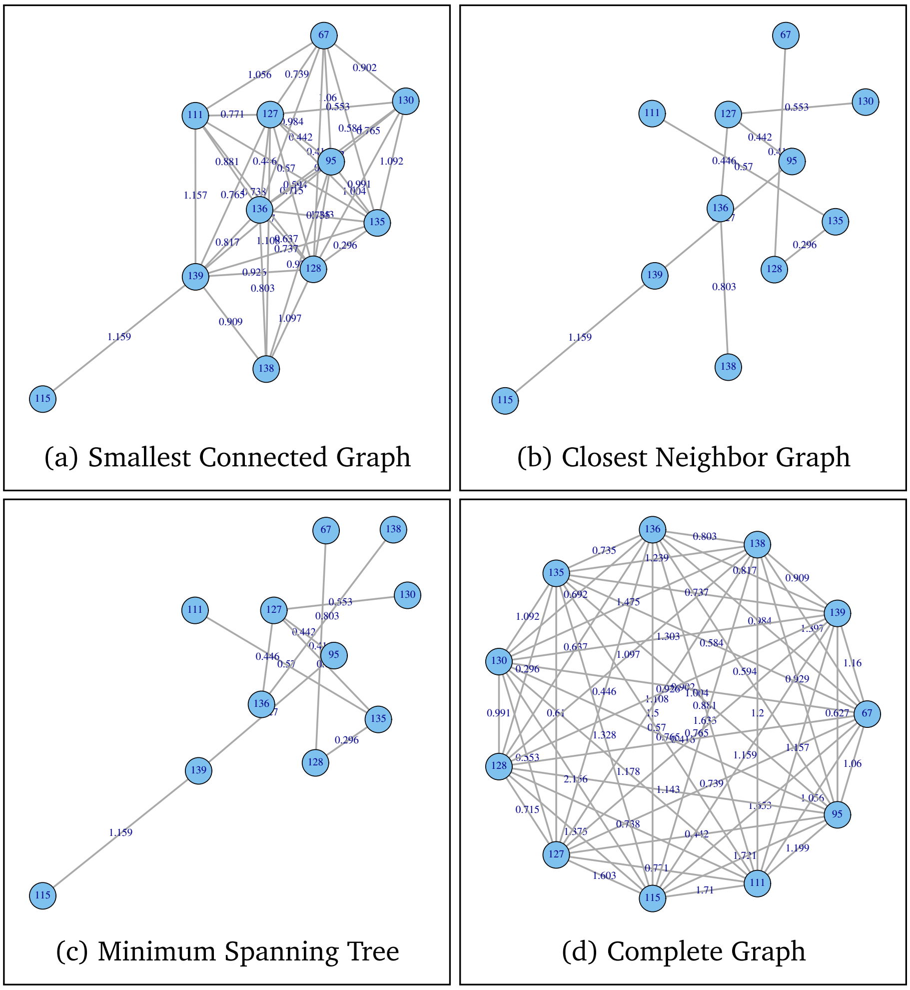
Our graph analytics tool follows a two phases work flow. In the first phase, the similarity graph is built and in the second phase, the built graph can be analyzed with different algorithms. The next paragraphs describe the work flow in more detail.

4.4.1 Building Similarity Graphs

In the first work flow phase, the similarity graph of RPs of the users of the web community has to be build. We use an undirected, vertices and edges labeled, weighted graph without loop to build a model for the similarity of the web community users. The weighted edges represent the similarity between the vertices, which stand for the users. The edges are labeled with the similarity value, that is the Manhattan Distance between the RPs of the users. The labels of the vertices are the user IDs. We use an undirected graph because the similarity of two users can be interpreted in both directions. Figure 4.1 shows examples of similarity graphs. As mentioned before, in the research field of social networks graph analysis is used to detect social structures between the users, like in [AA03]. These graphs represent the friend relationship of the users and are in comparison to our work different. We use *weighted* graphs to embody the similarity of users where the edge weights represent the similarity between the interests of the users. Therefore, we are not able to use the graph analytics algorithm tools from the social network analysis.

In our tool, the web site owner can choose different alternatives to build a similarity graph for the analysis. The vertices of the graph (the users) are connected via edges that represent the similarity. It is possible to connect every user to all other users so that a complete graph represents the similarity between all users. This graph is huge and not easy to understand. To reduce the complexity of this graph we introduce a *sim-*

[5] http://www.r-project.org/
[6] http://stat.bell-labs.com/S/
[7] http://www.r6rs.org/
[8] http://www.bioconductor.org/packages/release/bioc/html/Rgraphviz.html
[9] http://igraph.sourceforge.net/
[10] http://igraph.sourceforge.net/doc/html/index.html



(a) Smallest Connected Graph (b) Closest Neighbor Graph

(c) Minimum Spanning Tree (d) Complete Graph

Figure 4.1: Different types of similarity graphs

ilarity threshold. This threshold defines how similar the users must be to be connected together. Only users are connected via vertices whose Manhattan Distance of their RPs is smaller (remember: the smaller the distance is the more similar the users are) than the chosen threshold. Our analysis tool provides several predefined options to build different graphs with different thresholds. All these graphs are subgraphs of the complete similarity graph of the whole web community:

- **Smallest connected graph**: with this option the similarity threshold increases until every user has at least one connection to another user. In Figure 4.1a, user no. 115 was added last to the graph and has a Manhattan Distance of 1.159. Accordingly, all connected vertices have a similarity smaller or equal to 1.159. The result is *one* connected graph.

- **Closest neighbor graphs**: here users are only connected with their most similar neighbors. Every vertex has at least one edge to another vertex. If there exist more most similar neighbors with the same edge weight, the vertex is connected to all of them. This can result in *many* independent graphs as displayed in Figure 4.1b. The difference to the nearest neighbor algorithm is that the nearest neighbor algorithm calculates a path through an existing graph by choosing always the nearest neighbor of the actual vertex.
- **Minimum spanning tree** [Pri57]: a subgraph where all users are connected together with the most similar users. In contrast to the "closest neighbor graph", we have *one* connected graph. A minimum spanning tree is shown in Figure 4.1c.
- **Threshold graph**: at last the web site owner can choose a similarity threshold on her/his own. To simplify the choice, the tool suggests two thresholds to the owner: a minimum threshold and a maximum threshold. With the minimum threshold only the most similar users are connected together and with the maximum threshold all users are connected together with every user (complete graph). The owner can choose a value between the suggested thresholds to get meaningful results. Figure 4.1d shows a graph built with the maximum threshold.

4.4.2 Similarity Graph Mining Algorithms

In the second work flow phase, the web site owner can analyze the graph, generated in the first phase of the work flow, with different algorithms. The aim here is to detect the important users in the graph.

What is an important user? There exists not only one valid definition because it depends–as always–on the point of view. In social networks, e.g., the importance of users often stands for their reputation. The reputation of a user can be measured, e.g., by its number of connectors to other users. Therefore, a connector in social networks has another meaning, i.e., the friendship, like in our similarity graphs, we can not use this definition of user importance.

In a social graph a user could be important if she/he is central in respect to the graph. Centrality means that from this very user all other users should be not far away–it should be the nearest neighbor. These highly connected users are often referred as *Hubs* or *Authorities* [Kle99]. Hubs have many outgoing edges while Authorities have many incoming edges.

In a weighted similarity graph high importance could mean that this user is the most similar to other users–she/he should have many edges to other vertices and the edges weights should be as low as possible.

Accordingly, we provide nine algorithms to discover the importance of users. Therefore, the importance is defined by the used algorithm, which are explained below.

- **PageRank**: The vertex with the highest “PageRank” is the most important user.
- **Jaccard similarity coefficient**: We interpret the most similar vertex as the most important user.
- **Dice similarity coefficient**: Like above, we interpret the most similar vertex as the most important user.
- **Nearest neighbors degree**: If a vertex has many neighbors it can be considered as important.
- **Closeness centrality**: Vertices with a low closeness centrality value are important.
- **Hub score**: Vertices with a high hub score are named “hubs” and should be important.
- **Eigenvector centrality**: Vertices with a high eigenvector centrality score are considered as important users.

As these seven algorithms above are not *deliberately* designed to find the important vertices, i.e., users, in similarity graphs of user interests, we developed two new algorithms:

- **Weighted degree**: This simple algorithm chooses the vertex with the most connections. Vertices with many connections are important users because they are similar to other users. Actually, they are connected with other users cause of their similarity. If there are vertices with the same number of connections it takes the vertex with the lowest edge weights. Therefore, the most unimportant vertex has fewer connections to other vertices and the highest edge weights.
- **Range centrality**: The idea behind this algorithm is that a user is very important if she/he has many connections in comparison with the other users of the graph, if she/he has a short distance to her/his neighbors, and if her/his edges have low weights. Therefore, the range centrality is defined as follows:

Definition 4.4 *Range Centrality C_r:*

$$C_r = \frac{range^2}{aspl + aspw}$$

The *range* is the fraction of the number of users that are reachable from the analyzed vertex and of all users of the graph. We take the square of the range because we consider a user as very important that is connected with many other users:

Definition 4.5 *Range:*

$$range = \frac{\#reachable\ users}{\#all\ users}$$

The average shortest path length (*aspl*) is the average length of all shortest paths divided by the number of all shortest paths. The shortest paths are calculated with the analyzed vertex as starting point:

Definition 4.6 *Average shortest path length (*aspl*):*

$$aspl = \frac{average\ shortest\ path\ length}{\#shortest\ paths}$$

With the average shortest path weight (*aspw*) we take into account that the weight of the connected vertices should be very low, i.e., the vertices should be very similar. It's the fraction of the sum of all shortest paths weights and of the number of all shortest paths:

Definition 4.7 *Average shortest path weight (*aspw*):*

$$aspw = \frac{sum\ of\ all\ shortest\ path\ weights}{\#shortest\ paths}$$

In Chapter 5 we will use our analysis tool with real usage data and compare our new algorithms with the established ones.

4.5 Conclusion and Future Work

With the results of graph analysis we are now able to answer the research questions of Chapter 4:

- Which are the important users of the on-line community?
  We provide several algorithms (see Section 4.2.3) to calculate the important users of the community. The definition of importance is dependent on the used algorithm and on a subjective point of view. For example, vertices with many low weight connections can be considered as the important users of the community. These users are very similar to the other users, expressed by the low edge weight.

- Which users have similar interests?
  All users are connected via weighted edges. Users with similar interests have connections with low weights. The web site owner can also define, which users are connected together by selecting a similarity threshold (see work flow phase one, Section 4.4.1). As a result only similar users are connected via edges.

- How similar are the interests of the users of the on-line community?
  The weights of the edges of the similarity graph represent the similarity of the users. These weights are calculated with the Manhattan Distance. Therefore, the lower the weights of the edges are the more similar are the users of the community. We give the web site owner the possibility to set thresholds to identify quickly the similarity of her/his community (see Section 4.4.1).

- How is the community structured? Is it a homogeneous community where every user has similar interests or is it heterogeneous?
  The visualized graph of the community will give the web site owner an overview over the structure of the whole community of her/his web portal.

With answers to these questions, a web site owner is now able to start more focused marketing campaigns. To test new contents or features for her/his web site she/he could start with the most similar users because these users can be considered as an archetype for her/his community.

Besides the extension of the tool with more algorithms for the similarity calculation, in future, the exploration for similarity (or importance) metrics would be helpful. With this type of metrics it would be possible to evaluate the similarity algorithms objectively.

Table 4.1: Contribution summary

Subject	***Detail***
User Similarity	The interest of two users are equal, if their interest values of their RP is equal (Definition 4.1).
Similarity Threshold	The user similarity of a pair of users must meet a certain value to be connected together with an edge in a similarity graph (Section 4.4.1).
Similarity Graphs	We presented four different alternatives to build a similarity graph (Section 4.4.1): – *Smallest connected graph*: the similarity threshold increases until every user has at least one connection to another user. – *Closest neighbor graph*: users are only connected with their most similar neighbor(s). – *Minimum spanning tree*: all users are connected together with the most similar users. – *Threshold graph*: the web site owner can choose a similarity threshold and he can control how similar users must be to be connected in the similarity graph.
User Importance	We used seven common algorithms to calculate important users: PageRank, Jaccard similarity coefficient, Dice similarity coefficient, nearest neighbors degree, closeness centrality, hub score, and eigenvector centrality (Section 4.4.2)
	We presented two *new* algorithms to calculate the importance of a user: *weighted degree* (Section 4.4.2) and *range centrality* (Definition 4.4)

5 Case Studies

In this chapter, the framework analysis for managing the feedback of web site visitors (see Chapter 3) as well as the tool for building and mining similarity graphs (see Chapter 4) are validated with the data of real web site users. Three case studies with different focuses are conducted. The basic ideas of this chapter were partially published in the article of Schefels and Zicari [SZ12] and the paper of Schefels [Sch12].

5.1 Introduction

In Chapter 3 and 4, we introduced methods to analyze user feedback of on-line communities. In this chapter, we validate these concepts with three case studies with users of a real web site. We used the web site of the Databases and Information Systems (DBIS) research group at the Goethe-University of Frankfurt[1]. The web site is currently used as an information portal for both research and teaching related to databases and information systems. It has a public and a private section that is only accessible via a free registration. The Gugubarra Framework version 3.0 has been installed on the DBIS server and tracks the explicit and implicit feedbacks of all *registered* users.

The rest of this chapter is structured as follows: in Section 5.2 we conduct a case study that is focused on explicit user feedback. We examine the user data with our

[1] http://www.dbis.cs.uni-frankfurt.de/

framework analysis for managing the feedback in Section 5.2.2 and calculate the most important and unimportant users with the help of the tool for building and mining similarity graphs in Section 5.2.3. Section 5.2.4 presents the discussion of this case study.

In the next case study, presented in Section 5.3, the consistency of user behavior in a cold-start situation is compared with behavior in a warm-start situation. Furthermore, we used the framework analysis for managing the feedback, Section 5.3.2, and the tool for building and mining similarity graphs, Section 5.3.4, to analyze the on-line community. The results of this analysis are shown in Section 5.3.5.

The last case study, shown in Section 5.4, examines the data of all users of the DBIS web site with the framework analysis for managing the feedback (see Section 5.4.2) as well as with the tool for building and mining similarity graphs (Section 5.4.3). Section 5.4.4 sums up the conclusion of this case study, while Section 5.5 gives the overall conclusion regarding all case studies and outlines future work.

5.2 Case Study: Explicit Feedback

In the first case study, the feedback case study, we are interested in the explicit user feedback. We want to check how consistent the web site users are in their explicit feedback. Therefore, we compared the first feedback of a new web site user at her/his first login with the feedback, shortly given before she/he logs out for the first time from the web site.

5.2.1 System Settings

The feedback case study was performed as follows: we asked a number of students to register to the web site and perform a number of tasks related to its content. We computed for each registered user a NOP and asked for an explicit feedback twice via a web-based form. The explicit feedback was requested first at the beginning of the tasks and secondly at the end of the tasks. After that, we computed an RP for each user who performed the explicit feedback case study. We assumed a *cold-start situation* [ME95], where users had to register to the web site for the first time and no previous information of the users was known. We used the following settings:

a) NOP with parameter $a = b = 0.5$:
For the *cold-start*, we considered for this test, the time and the activity performed by each user equally important in the calculation of the NOP.

b) We defined four topics for the web site (see Section 3.3, step 1):
$T_1 = teaching$, $T_2 = research$, $T_3 = databases$, and $T_4 = news$.

c) RP filter function constant with $f_l(S_k) = 1$ (see Section 3.4, step 2).
For the cold-start, we considered user explicit feedback with the same importance as a NOP. This is justified by the fact that no previous information about the users was given and, therefore, in the *cold-start* all inputs were considered equally important.

For each topic, zone topic weights were associated with different *zones* [HKTZ06b]. Next, we followed the seven steps of our framework, see Chapter 3, to analyze the feedback of the test participants.

5.2.2 Framework Analysis for Managing Feedback of Visitors of a Web Site

In Chapter 3, we introduced our framework for the analysis of the feedback of web site users. With the data from this feedback case study we will test our concept. The framework consists of seven steps, which will be performed in the next paragraphs.

Step 1: Definition of a scope. First, we define the scope S_k for our feedback case study: we only wanted to include the participants of the study into the analysis, and not all visitors of our web site. This means that the cluster of users consisted of eleven web site users:

$$\begin{aligned} Cluster_{u_m} = \{&user_{67}, user_{95}, user_{111}, user_{115}, user_{127}, \\ &user_{128}, user_{130}, user_{135}, user_{136}, user_{138}, user_{139}\} \end{aligned}$$

The cluster of topics, we liked to analyze, comprises the four topics:

$$Cluster_{T_i} = \{teaching, research, databases, news\}$$

For these topics we asked the users for their feedback (see also Figure 5.2).

Step 2: Definition of a filter. We assume a *cold-start situation* [ME95], where users had to register to the web site for the first time and no previous information of them

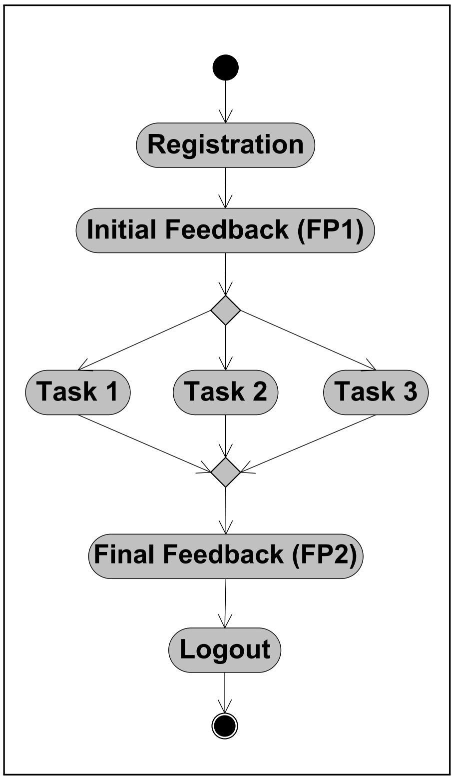


Figure 5.1: Activity diagram of the feedback case study

was known. For a *cold-start* we wanted the full impact of the explicit feedback on the RP calculation. Therefore, we used a constant filter function:

$$f(Cluster_{u_m}, Cluster_{T_i}) = 1$$

With this function the explicit feedback of all users in the $Cluster_{u_m}$ has the same importance.

Step 3: Obtaining explicit user feedback. Eleven students of computer science were asked to take part in the test, which took place on 2010-12-15. For each of the eleven students one user session was analyzed. See Figure 5.1 for the flow chart of the experiment. At registration time, we asked each new student to perform a set of tasks related to the four topics defined in step 1. The students were asked to:

1. First, give an initial explicit feedback on their interests with respect to the four topics.

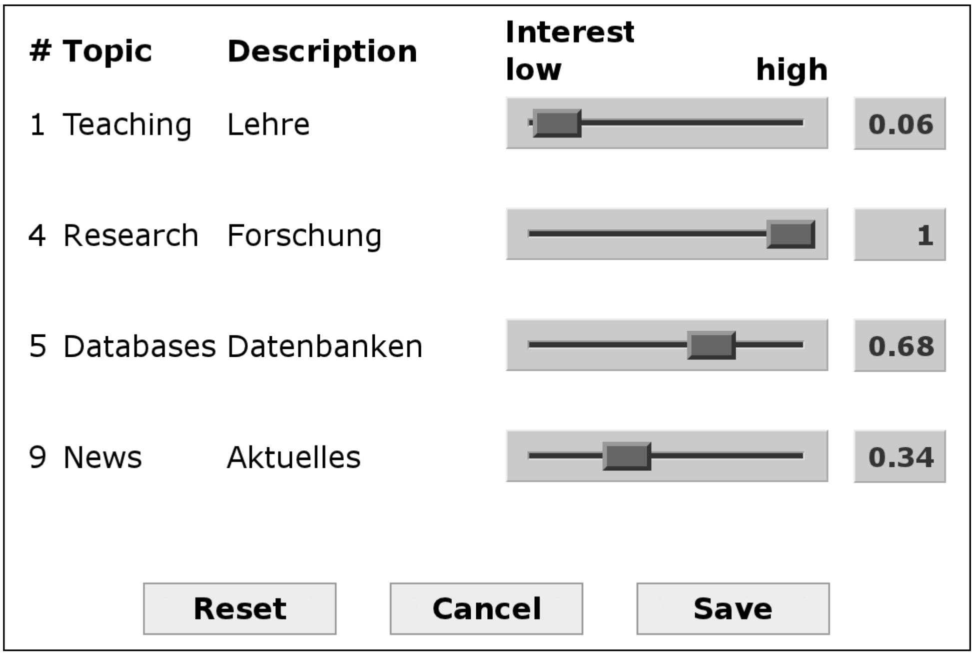


Figure 5.2: Screenshot of the Gugubarra feedback form

2. Second, choose among a set of predefined tasks related with $T_1 = teaching$, $T_2 = research$, $T_3 = databases$, and $T_4 = news$. The participants were able to execute one or more tasks.

3. At the end, after performing the selected tasks, the students were asked to give an additional feedback. The students were shown their initial feedback and were asked to update it, if their interest had changed after conducting the task or tasks they had selected.

The invitation for the test, including the tasks to be performed by the students, is illustrated in Section A.1.1 in Appendix A. Thereby, a student could give a feedback by simply using a web based slider (see Figure 5.2), which for each topic listed could span from 0 (no interest) to 1 (maximum interest). The students had the chance to review their last feedback.

Step 4: Filtering the user feedback. In this step of our framework, we apply the filter function f_l, which was defined in step 2, on the scope S_k identified in step 1. Therefore, we calculated with Formula 3.2 the RP for all users that belong to the scope S_k. First, we needed to calculate the NOP with Formula 2.3 per user and write the explicit feedback into the FP of each user. Second, we calculated the RP. The formulas below show, as an example, the calculation of the RP for the specific user no. 127.

The NOP and the FP of the $user_{127}$ with $t_0 = $ 2010-12-20:

$$NOP_{u_{127},t_0}\begin{pmatrix} teaching \\ research \\ databases \\ news \end{pmatrix} = \begin{pmatrix} 0.29 \\ 0.27 \\ 0.21 \\ 0.27 \end{pmatrix}$$

$$FP_{u_{127},t_0}\begin{pmatrix} teaching \\ research \\ databases \\ news \end{pmatrix} = \begin{pmatrix} 0.71 \\ 0.77 \\ 0.82 \\ 0.77 \end{pmatrix}$$

The RP of the $user_{127}$:

$$RP_{u_{127},t_0}\begin{pmatrix} teaching \\ research \\ databases \\ news \end{pmatrix} = \frac{NOP_{u_{127},t_0}\begin{pmatrix} teaching \\ research \\ databases \\ news \end{pmatrix} + f_l(S_k) * FP_{u_{127},t_0}\begin{pmatrix} teaching \\ research \\ databases \\ news \end{pmatrix}}{a + b + f_l(S_k)} =$$

$$= \frac{\begin{pmatrix} 0.29 \\ 0.27 \\ 0.21 \\ 0.27 \end{pmatrix} + 1 * \begin{pmatrix} 0.71 \\ 0.77 \\ 0.82 \\ 0.77 \end{pmatrix}}{0.5 + 0.5 + 1.0} = \begin{pmatrix} 0.50 \\ 0.52 \\ 0.51 \\ 0.52 \end{pmatrix}$$

Step 5: Clustering. In general, clustering the RPs facilitates the interpretation for the owner of web site of large on-line communities (i.e., with high number of users). RPs can be clustered together in several different ways, for example, grouping together users with RPs showing similar behavioral patterns of interest. In this feedback case study, it was not necessary to cluster because we only had eleven users.

Step 6: Consistency check. To check the consistency of the users, we projected our metric interest scale on an ordinal scale. We used an ordinal scale with three ordinal categories:

$$
\begin{aligned}
D = \{&``low\ interest", ``medium\ interest", ``high\ interest"\} \text{ with}\\
&``low\ interest" = [0.0, 0.3),\\
&``medium\ interest" = [0.3, 0.7), \text{ and}\\
&``high\ interest" = [0.7, 1.0]
\end{aligned}
$$

Next, we looked at each single user whether her/his FP or NOP and RP is located into the same category of the ordinal interest scale. This case study focuses on the explicit feedback of the web users. Therefore, we compared the consistency of the initial feedback (FP_1) with the consistency of the second feedback (FP_2) of each user per topic.

Figures A.1 to A.4 in Appendix A, display an RP as red circle, a NOP as green diamond, the first FP as a blue plus, and the second FP as a blue "x", for each user per topic. If the first and the second FP have the same values a blue star is drawn. The x-axis displays both the interest value of the users in metric form and the ordinal categories. On the y-axis the user numbers are depicted.

The results of the consistency check for participants of the feedback case study are summarized in Table 5.1. The table displays the type of consistency, e.g., $C^{e\downarrow}$, of the users for both feedbacks (columns "FP_1" and "FP_2") and the ordinal category (column "*Cat.*", $med = medium$) where the profiles of the consistent users are located.

When we applied the consistency check for the "*teaching*" topic we obtained the results depicted in Figure A.1, shown in Appendix A. The results are summarized in Table 5.1, where we can see that four users (about 36% of the users) are *total inconsistent* ($\not{C}^t$) in both FPs, while two users (18%) are *total consistent* (C^t), both located in the low interest category. Two users are *down explicit consistent* ($C^{e\downarrow}$) in both FPs, one user (9%) is *up explicit consistent* ($C^{e\uparrow}$), and another chances from *down explicit consistent* ($C^{e\downarrow}$) to *total inconsistent* ($\not{C}^t$). At last, one user is *up implicit consistent* ($C^{i\uparrow}$) in the first FP and *down implicit consistent* ($C^{i\downarrow}$) in the second FP.

The results of the consistency check for the "*research*" topic are presented in Figure A.2 (in Appendix A), and summarized in Table 5.1. The table shows that in the "research" topic five of the users (45%) are *total inconsistent* and only one (9%) is *total consistent* in the low interest category. Two users are *down explicit consistent* and one is *up explicit consistent*. Two users are changing their consistency level: one from *up implicit consistent* to *down implicit consistent* and the other from *up implicit consistent* to *total consistent*.

When the consistency check was applied for the "*databases*" topic (see Figure A.3 in Appendix A), the following results were obtained (Table 5.1). In this topic, we have three *total inconsistent* users (27%) and three *total consistent* users. In the *down explicit*

Table 5.1: Result of the consistency check for the feedback case study

	Topic											
	Teaching			**Research**			**Databases**			**News**		
user$_{\textbf{id}}$	FP_1	FP_2	Cat.	FP_1	FP_2	Cat.	FP_1	FP_2	Cat.	FP_1	FP_2	Cat.
$user_{67}$	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-	$C^{i\uparrow}$	C^t	low
$user_{95}$	C^t	C^t	low	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-
$user_{111}$	$C^{e\downarrow}$	$C^{e\downarrow}$	high	$C^{e\downarrow}$	$\not{C}^t$	med	$C^{e\uparrow}$	$C^{e\uparrow}$	low	$\not{C}^t$	$C^{e\downarrow}$	med
$user_{115}$	$C^{i\uparrow}$	$C^{i\downarrow}$	med	$C^{i\uparrow}$	C^t	low	$C^{i\uparrow}$	C^t	low	C^t	C^t	low
$user_{127}$	$C^{e\downarrow}$	$\not{C}^t$	med	$C^{e\downarrow}$	$\not{C}^t$	med	$\not{C}^t$	$\not{C}^t$	-	$C^{e\downarrow}$	$\not{C}^t$	med
$user_{128}$	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-	$C^{i\uparrow}$	$C^{i\downarrow}$	med	C^t	C^t	low
$user_{130}$	$\not{C}^t$	$\not{C}^t$	-	$\not{C}^t$	$\not{C}^t$	-	C^t	C^t	high	$\not{C}^t$	$\not{C}^t$	-
$user_{135}$	$\not{C}^t$	$\not{C}^t$	-	$C^{e\downarrow}$	$C^{e\downarrow}$	med	C^t	C^t	med	$C^{i\uparrow}$	$C^{i\uparrow}$	low
$user_{136}$	$C^{e\downarrow}$	$C^{e\downarrow}$	med	$C^{e\downarrow}$	$C^{e\downarrow}$	med	$C^{e\downarrow}$	$C^{e\downarrow}$	med	$C^{i\uparrow}$	$C^{i\uparrow}$	med
$user_{138}$	C^t	C^t	low	$\not{C}^t$	$\not{C}^t$	-	C^t	C^t	low	$\not{C}^t$	$C^{e\downarrow}$	med
$user_{139}$	$C^{e\uparrow}$	$C^{e\uparrow}$	low	C^t	C^t	low	$C^{e\downarrow}$	$C^{e\downarrow}$	med	$C^{i\uparrow}$	C^t	low

consistent level are two users, in the *up explicit consistent* level only one. Two users are *up implicit consistent*; one changes to *total consistent* and the other to *down implicit consistent*.

Next, we apply the consistency check for the "*news*" topic (see Figure A.4 in Appendix A). The following results were obtained (Table 5.1): we have two *total inconsistent* users (about 18% of the users) and two *total inconsistent* who change to *down explicit consistent*. *Total consistent* are two users, with one *down explicit consistent*, who changes to *total inconsistent*. The rest of the users, four, is *up implicit consistent* in the first FP. Two stay in this consistency level while the others change to *total consistent*.

Step 7: Interpreting the results of the consistency check. Within this step, we interpret the results from the consistency check. In this case study, we focus on the explicit feedback of the participants. The *explicit consistency* shows, whether the explicit feedback of the users is consistent, i.e., is in the same interest category like the RP. The direction of the *implicit consistency* (up or down) indicates, whether the implicit feedback over- (up) or underestimates (down) the interests of the users. This means in detail, that for an *up implicit consistent* user the FP is in a higher interest category as the NOP and the RP. For a *down implicit consistent* user it is the exact opposite.

The first FP was given by the users before they explored the web pages while solving the tasks of the experiment. Therefore, the first FP represents their expectations of the web site. The second FP, given at the end of the experiment, reflects their web site experience during the experiment and how this experience meets their expectations. Next, we interpret the results from the consistency check topic by topic.

Topic "teaching"

Measures: Only two users changed their consistency, one from *up implicit consistent* to *down implicit consistent*, the other one from *down explicit consistent* to *total inconsistent*. All other users remained in the same consistency level, most of them were *total inconsistent*.

Interpretation: On the one hand, there were three *explicit consistent* users, which indicates that the users were interested in the topic (explicit feedback), but did not visit many pages within the "teaching" topic. This is indicated by the fact that there were two *total consistent* users. On the other hand, the high percentage of *inconsistent* users indicates that the web pages with the "teaching" topic do not meet the expectations of the visitors because all interest profiles were in different *interest levels*. Thus, the participants of our study were students of computer science, they know the teaching pages of our web site already very well and could solve the tasks of the study without visiting the related web pages. It is also possible that only few people selected a task related to the "teaching" topic.

Topic "research"

Measures: We observed that two users switched from *down explicit consistent* to *total inconsistent* and one from *up implicit consistent* to *total consistent*.

Interpretation: It seems, cause of the many *total inconsistent* users, that the users had different expectations of the pages with the "research" topic. It may also be possible that the research part of the web site is not well structured, so that the users did not find what they were searching for. Another cause could be that the tasks, which were related to the "research" topic, were too difficult, so that no user did perform any. That would also explain why the NOP of the users was always in the lowest interest category. In detail, many *down explicit consistent* users were in the medium interest category. According to the definition of *down explicit consistent*, their NOPs must be located in the low interest category because their FP and RP were in the medium interest category.

Topic “databases”

Measures: One *up implicit consistent* user changed to *total consistent* and another *up implicit consistent* user to *down implicit consistent*.

Interpretation: In the view of the high number of *total consistent* users, this topic seems to be interesting for the users and the content of the pages measure up to their expectations. Nevertheless, there were two users with a FP in a higher interest category than their NOP. This means, that we still have potential to adopt the content related to this topic to meet the expectations of the users.

Topic “news”

Measures: Two *implicit consistent* users changed to *total consistent*, two *total inconsistent* users to *down explicit consistent*, and one *down explicit consistent* to *total inconsistent*.

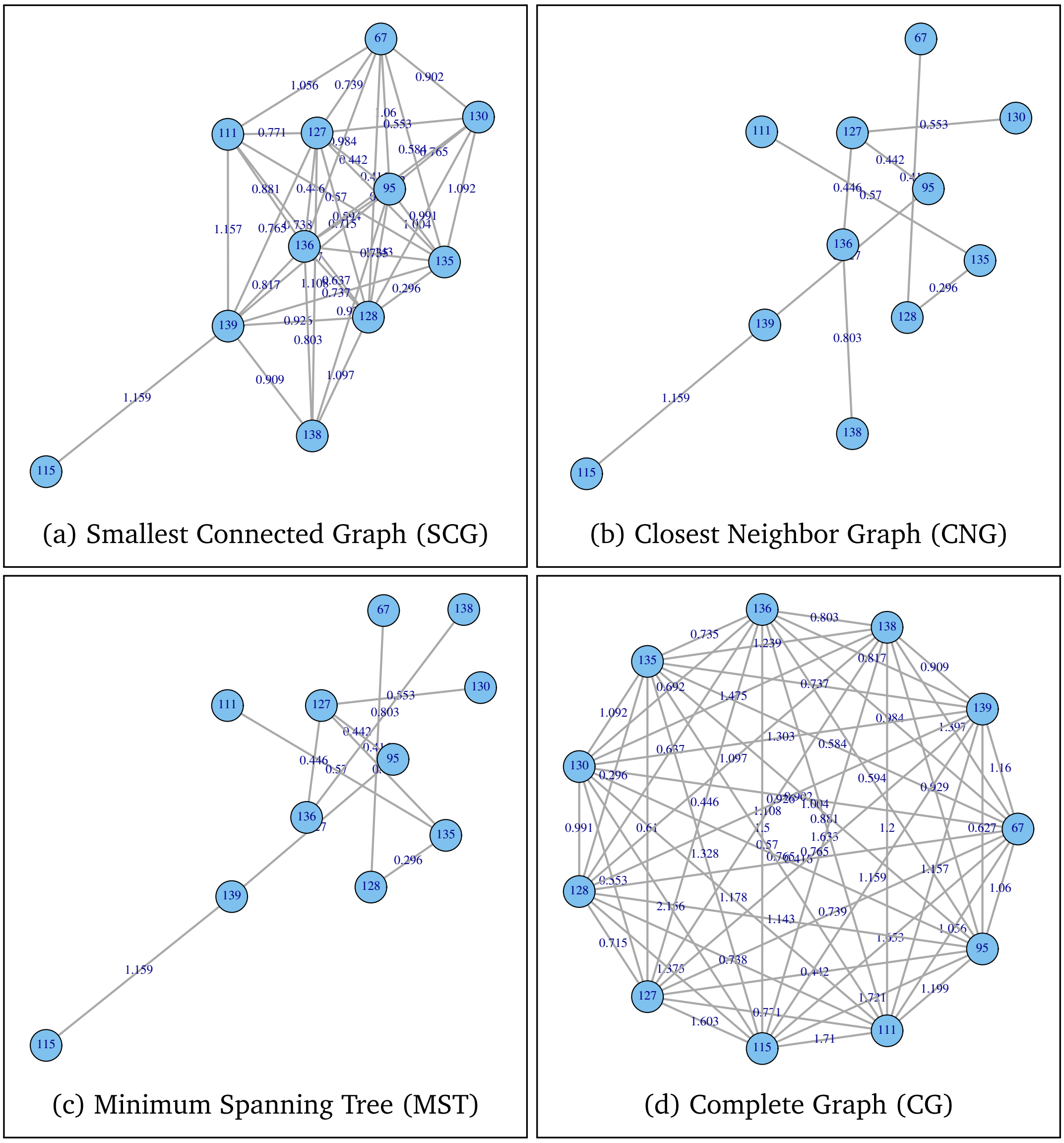
Interpretation: In this topic, the users switched the consistency level most frequently. From the view of the web site owner, this is a positive sign because here the users changed mostly into a more “strict” consistency, i.e., from C^i to C^t or from $\not{C}^t$ to $C^{e\downarrow}$. This means, that the users become more interested in the topic while visiting the web site. Another explanation could be that the web site originates from a databases and information systems research group, and, therefore, most of the news items are related to databases. And, as shown above, the users were very interested in the “databases” topic.

All Topics

We also checked the consistency of the users for all topics. The following users presented a consistent behavior:

- $user_{95}$: this user was three times *total inconsistent* and one time *total consistent* (“teaching” topic). She/he seems to be only interested in the “teaching” topic.
- $user_{115}$: this user switched three times the consistency and was three times *total consistent* with her/his second feedback.
- $user_{130}$: this user was three times *total inconsistent* and one time *total consistent* in the “databases” topic. She/he seems to be only interested in this topic.

If the web site owner modifies web pages with the “teaching” or “databases” topic, a change in the consistent behavior of $user_{95}$ and $user_{130}$ might be a sign that the changes to the web site are effective or not.



(a) Smallest Connected Graph (SCG)

(b) Closest Neighbor Graph (CNG)

(c) Minimum Spanning Tree (MST)

(d) Complete Graph (CG)

Figure 5.3: Similarity graphs of the feedback case study

5.2.3 User Similarity

Next, we analyzed the users with the help of similarity graphs as described in Chapter 4. The aim here is to detect important users. The importance of a single user is determined by the different algorithms that were introduced in Section 4.4.2.

First phase. In the first phase of the analysis process, we generated the similarity graphs of the users. These four graphs are displayed in Figure 5.3.

Table 5.2: Evaluation results: IDs of the users with maximum and minimum importance of every graph type (rows) for different algorithms (columns)

		Page Rank	Nearest N.D.	Dice S.C.	Jaccard S.C.	Closen. C.	Hub Score	Eigen-vector C.	Weight. Degree	Range C.
SCG	Max	127	135	135	135	127,128,136	127,128,136	128	127	127
	Min	115	139	115	115	115	115	115	115	115
CNG	Max	127	95	95	95	127	127	139	127	127
	Min	130	139	139	139	67,111	128,135	111	115	111
MST	Max	127	130	95	95	127	127	139	127	127
	Min	115	139	139	139	115	115	67	115	115
CG	Max	127	127	all	all	all	all	115	127	127
	Min	115	115	all	all	all	all	127	115	115

Second phase. In the second phase, the we analyzed the graphs, generated in the first phase, with different algorithms. The aim here is to detect the important users of the on-line community.

Table 5.2 displays the results of our calculations. The rows present the different graph types: *SCG* stands for *s*mallest *c*onnection *g*raph, *CNG* for *c*losest *n*eighbor *g*raph, *MST* for *m*inimum *s*panning *t*ree, and *CG* for *c*omplete *g*raph (i.e., threshold graph). For each graph type, the user/users with maximum and minimum importance is displayed. Every column presents one importance algorithm. We can observe the following fact in the dataset in respect to our new algorithms, the weighted degree and the rang centrality:

In the SCG, the user no. 127 is calculated as most *important* user by the range centrality, weighted degree, hub score, closeness centrality, and the PageRank algorithm. The Dice similarity coefficient, the Jaccard similarity coefficient, and the nearest neighbor degree calculate user no. 135 as most *important* and the eigenvector centrality user no. 128. So, the majority of the algorithms select user no. 127 as most *important*. The most *unimportant* user is user no. 115, calculated by all algorithms except the nearest neighbor degree, which chooses user no. 139.

In the CNG, the user no. 127, like in the SCG, is calculated as most *important* user by range centrality, weighted degree, hub score, closeness centrality, and the PageRank algorithm. The Dice similarity coefficient, the Jaccard similarity, and the nearest neighbor degree calculates user no. 95 as most *important*. Only the eigenvector centrality selects user no. 129 as most *important*. Both, user no. 139 and user no. 111, are calculated by three algorithms as most *unimportant*.

In the MST, range centrality, weighted degree, hub score, closeness centrality, and the PageRank algorithm select user no. 127 as most *important*, the Dice similarity coefficient, the Jaccard similarity coefficient user no. 95. The nearest neighbor degree (user no. 130) and the eigenvector centrality (user no. 139) select different *important* users. User no. 115 is calculated by five algorithms as most *unimportant* user, user no. 139 by three algorithms, and user no. 67 only by one.

In the CG, user no. 127 is most *important* for four algorithms, which select also user no. 115 as the most *unimportant* user. The eigenvector centrality calculates user no. 115 as most *important* and user no. 127 as most *unimportant*–exact the opposite. The Dice similarity coefficient, the Jaccard similarity coefficient, the closeness centrality, and the hub score are not able to find an *un-/important* user in the complete graph because these algorithms do not include the edge weights into their calculation.

5.2.4 Discussion

This case study focused on *explicit user feedback*. We compared two explicit feedbacks of users during their first web site experience. The first feedback was given before the users visited the web site for the first time, the second afterwards. With the framework analysis, introduced in Chapter 3, we were able to detect weak topics of the web site in respect to a *cold-start situation*. We draw the following conclusion:

In both FPs, *total consistent* users indicate that the web pages meet the expectations of these users. In the feedback case study the total consistent users never change to another consistency level.

Total inconsistent users or a change to *total inconsistent* indicate that the web site does not meet the expectations of the users because their implicit feedback differs from their explicit feedback. The users can only change from total inconsistent to explicit consistent because only the FP changes in our feedback case study, whereas the NOP and the RP of a user are constant. In detail, a change from total inconsistent to *up explicit consistent* indicates that the users like the topics more after their web site experience. Concluded from their up explicit consistency (NOP in a higher ordinal category than the RP and FP), they performed many actions and stayed a long time on page and changed afterwards their FP to the same ordinal category as the RP; a change into a higher ordinal category. A change from total inconsistent to *down explicit consistent*, FP and RP are in a higher ordinal category than the NOP, indicates that the users dislike the web site, which may be caused by a bad web site experience. The FP changes from a higher ordinal category to a lower one and indicates that the users may have lost interest in the topic.

Implicit or *explicit consistent* users that change to *total consistent* denote that the users adapt the FP to their observed behavior (NOP) after their web site experience. In detail, a change from *down implicit* or *up explicit consistent* to total consistent indicates that they like the topic after their web visit more, which evinces a positive web experience. This conclusion can be drawn because such a change means that the FP has to change into a higher ordinal category, seen from the consistency level. A change from *up implicit* or *down explicit consistent* to total consistent could indicate that the web pages with this topic did not meet their expectation. Here, the FP changes to a lower ordinal category.

With the results of the consistency check, the web site owner is now able to identify weak topics and can adapt the web site accordingly. Before deploying the new web pages it would be wise to test these pages with the most important, i.e., most similar users of the case study. The analysis tool for building and mining similarity graphs detects in the explicit feedback study the most important users. The two new algorithms proved to be a good alternative to the common algorithms and calculate reasonable results.

5.3 Case Study: Cold-Warm-Start

In this section, we compare a *cold-start situation* with a *warm-start situation*. In a *cold-start situation*, no information about the interests of a user is known. This situation is typical for a on-line community, where a user registers and does the first login to the web site. In this situation, only the initial and empty user interest profiles are available. After the first user session (the first login and logout) the system enters the *warm-start situation*. Now, the user interest profiles are calculated from the data of the first session and can be compared with the second user session.

5.3.1 System Settings

In this section, we performed a case study to compare cold- and warm-start situations. We used the same settings for the Gugubarra Framework like in the explicit user feedback case study in Section 5.2.1:

a) NOP with parameter $a = b = 0.5$:
We considered for this test the time and the activity performed by each user equally important in the calculation of the NOP.

b) We defined four topics for the web site (see Section 3.3, step 1):
$T_1 = teaching$, $T_2 = research$, $T_3 = databases$, and $T_4 = news$.

c) RP filter function constant with $f_l(S_k) = 1$ (see Section 3.4, step 2).
We considered user feedback with the same importance as a NOP. Like in the last

case study, we had no previous information about the users. Therefore, all inputs had the same impact in the RP calculation.

For each topic, zone topic weights were associated with different *zones* [HKTZ06b].

In the next sections, we do a framework analysis, introduced in Chapter 3, of the collected data and search for the most important user using the graph analytics tool of Section 4.3.

5.3.2 Framework Analysis for Managing Feedback of Visitors of a Web Site

Step 1: Definition of a scope. First, we defined the scope S_k for the cold-warm-start case study: we only included participants into the analysis, which performed the complete experiment. This means, that the cluster of users consisted of eleven users:

$$Cluster_{u_m} = \{user_{250}, user_{252}, user_{255}, user_{259}, user_{261}, user_{266}\}, \\ user_{267}, user_{268}, user_{270}, user_{271}, user_{272}\}$$

The cluster of topics we liked to analyze comprises of the four topics:

$$Cluster_{T_i} = \{teaching, research, databases, news\}$$

For these topics we asked the users for their feedback.

Step 2: Definition of a filter. We assume a *cold-start situation* [ME95], where users had to register to the web site for the first time, and no previous information of the users was known. For a *cold-start*, we wanted the full impact of the explicit feedback on the RP calculation. Therefore, we used a constant filter function:

$$f(Cluster_{u_m}, Cluster_{T_i}) = 1$$

With this function, the explicit feedback of all users in the $Cluster_{u_m}$ has the same importance.

Step 3: Obtaining explicit user feedback. The cold-warm-start case study consisted of a start web page with a list of links to four web pages, see Figure 5.4. Every link had a short description, approximately one sentence, concerning the content of the linked page. So, the topic of the linked page was obvious to the participants. The four topics were: teaching, research, databases, and news ($Cluster_{T_i}$). The linked page contained a

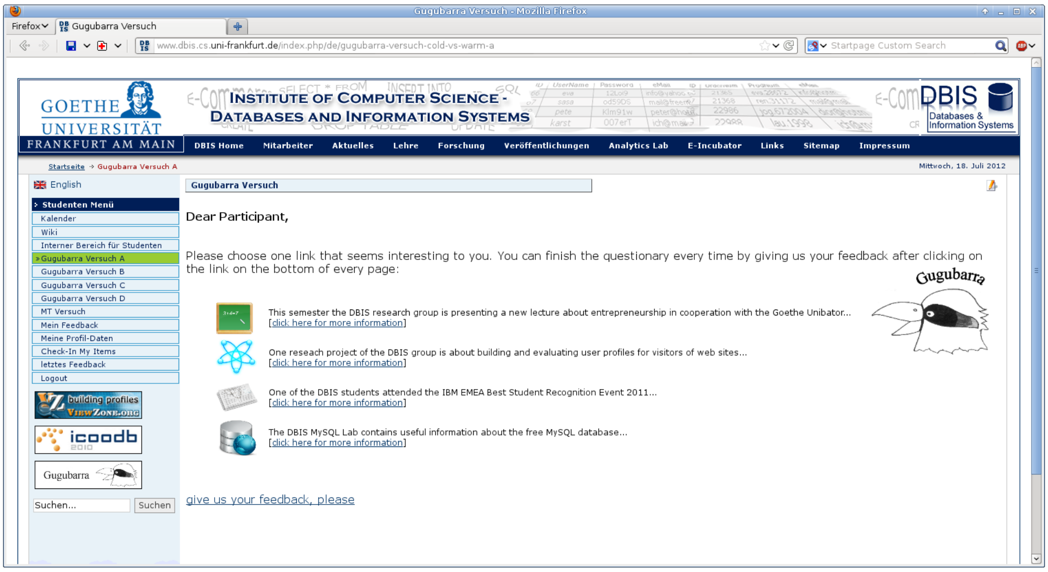


Figure 5.4: Screenshot of the start page

more detailed preview of a page of the web site related to the topic and the participants were able to click on a link in order to see the complete page. On every page the participants had the possibility to give their explicit feedback, see Figure 5.2. After the participants gave their feedback, the experiment ended. Figure 5.5 displays the flow chart of the cold-warm-start case study.

There were four different versions of the start page: two with a different order of the link list and two with different linked pages. The participants were uniformly distributed to the pages.

The case study was built up of two parts: the cold-start and the warm-start situation. In the cold-start situation, we had no information about the interests of the participants, all started with empty user profiles. In the warm-start situation, a user profile of each participants was calculated from the data of the cold-start situation.

Cold-start situation: Every participant received an invitation (see Section A.2.1) with the request to take part in the cold-warm-start case study. First, she/he had to register to the DBIS web site, where the Gugubarra Framework is installed. After the registration process, the participant was automatically led to one of the start pages with the list of links. Next, the participant was asked to follow one of the links to the page with the topic she/he was most interested in. The Gugubarra Framework monitored every action of the participant and built an initial user profile from these information after the user session.

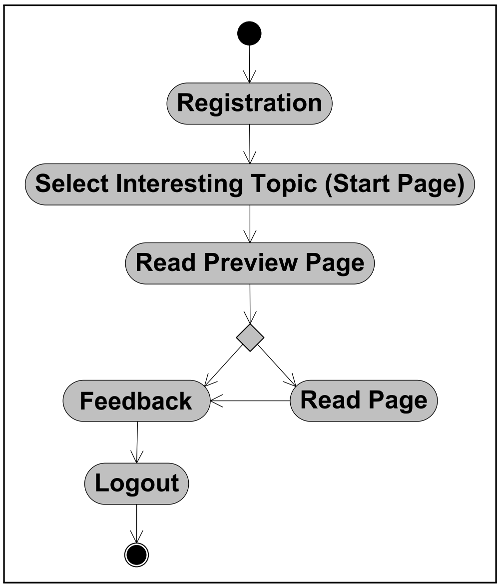


Figure 5.5: Activity diagram of the cold-warm-start case study

Warm-start situation: After one week, all participants were asked to take part in the same experiment a second time. This time, the participants started with a start page with a different order of the link-list and with different linked pages. Additionally, three participants conducted the experiment in a supervised lab condition. These three participants performed two warm-start situations. However, eleven participants accomplished the complete cold-warm-start case study.

A total of 19 participants took part in this experiment. Three performed the experiment supervised in a lab, 16 unsupervised on-line via Internet connection. Seven of the 16 on-line participants scheduled the complete experiment with its two parts while eight only completed one of the two parts. One participant made an additional second warm-start. The participants were all members or students of the computer science department of the Goethe-University Frankfurt, nine of them were female (39%) and 14 male (61%). The first part of the experiment was conducted at 2011-12-12, the second at 2011-12-19.

Step 4: Filtering the user feedback. In this step of our framework, we apply the filter function f_l, defined in step 2, on the scope S_k, identified in step 1. Therefore, we calculated with Formula 3.2 the RP for all users of the scope. First, we needed to calculate the NOP with Formula 2.3 per user and write the feedback into the FP of each user. Second, we calculated the RP. In Section 5.2.2, an example of the calculations is shown.

Step 5: Clustering. The clustering of the user RPs is only worthwhile for a large number of users. Because only 19 users conducted the experiment, it was not necessary to cluster their RPs.

Step 6: Consistency check. For the consistency check we used an ordinal scale with three ordinal categories:

$$\begin{aligned} D = \{&\text{“}low\ interest\text{''}, \text{“}medium\ interest\text{''}, \text{“}high\ interest\text{''}\} \text{ with} \\ &\text{“}low\ interest\text{''} = [0.0, 0.3), \\ &\text{“}medium\ interest\text{''} = [0.3, 0.7), \text{ and} \\ &\text{“}high\ interest\text{''} = [0.7, 1.0] \end{aligned}$$

Next, we looked at each single user whether her/his FP or NOP is located into the same category of the ordinal interest scale as the RP. We did a consistency check on the NOPs, FPs, and RPs of each user from the cold-start situation as well as from the warm-start situation, as defined in Section 3.8. The results of the consistency check are presented in Table 5.3. The consistencies (e.g., $C^{e\downarrow}$) of the cold-start are displayed in the "*Cold*" titled column, the warm-start consistencies in the "*Warm*" titled column. The interest categories are listed in the column "*Cat.*", separated by a "/" (slash) with the cold-start category in first place and the warm-start category in second place (*med*=*medium*, *hi*=*high*). If the consistency stays in the same category, it is displayed only once. Further, we compare the consistency of the *cold-start* situation (no information about the user's interest) with the consistency of the *warm-start* situation of each user per topic.

Figures A.5 to A.8 in Appendix A, display the first RP as red solid circles, the second RP as red circle, the first NOP as green diamond, the second NOP as green solid circle, the first FP as a blue plus, and the second FP as a blue "x", for each user per topic. If the first and the second FP have the same values a blue star is drawn. The x-axis displays both the interest value of the users in metric form and the ordinal categories. On the y-axis the user numbers are depicted.

The consistency check (see Table 5.3) for the "*teaching*" topic shows five users with a constant consistency: three *total consistent* (C^t), one *up implicit consistent* ($C^{i\uparrow}$), and one *down explicit consistent* ($C^{e\downarrow}$). In the cold-start situation three users are *up implicit consistent* ($C^{i\uparrow}$) users, two *down explicit consistent* ($C^{e\downarrow}$), and one *total consistent* (C^t). In the warm-start situation, we have four *explicit consistent* (C^e) users, three *down* ($C^{e\downarrow}$) and one *up* ($C^{e\uparrow}$).

In the "*research*", topic we observe six users who stay in the same consistency level: two *total consistent*, three *total inconsistent*, and one *up explicit consistent*. In the cold-start situation, three users are *total consistent*, one *total inconsistent*, and one *up implicit*

Table 5.3: Result of the consistency check for the cold-warm-start case study

	Topic											
	Teaching			**Research**			**Databases**			**News**		
user$_{\mathbf{id}}$	Cold	Warm	Cat.	Cold	Warm	Cat.	Cold	Warm	Cat.	Cold	Warm	Cat.
$user_{250}$	$C^{i\uparrow}$	$C^{e\downarrow}$	low/med	$C^{i\uparrow}$	$C^{e\downarrow}$	med/hi	$C^{i\uparrow}$	$C^{i\uparrow}$	low	$C^{i\uparrow}$	$C^{e\downarrow}$	low/med
$user_{252}$	C^{t}	$C^{e\downarrow}$	med/hi	C^{t}	C^{t}	low	$\not{C}^{t}$	$\not{C}^{t}$	-	C^{t}	C^{t}	low
$user_{255}$	$C^{e\downarrow}$	$C^{e\downarrow}$	hi	$\not{C}^{t}$	$\not{C}^{t}$	-	$C^{i\uparrow}$	$C^{i\uparrow}$	low	$\not{C}^{t}$	$\not{C}^{t}$	-
$user_{259}$	C^{t}	C^{t}	low	C^{t}	C^{t}	low	C^{t}	C^{t}	low	C^{t}	C^{t}	low
$user_{261}$	$C^{i\uparrow}$	$C^{e\downarrow}$	med	C^{t}	$C^{e\downarrow}$	med/hi	$C^{i\uparrow}$	$C^{i\uparrow}$	low	$C^{i\uparrow}$	$C^{i\uparrow}$	low
$user_{266}$	$C^{i\uparrow}$	C^{t}	low	$C^{e\uparrow}$	$C^{e\uparrow}$	low	$C^{e\downarrow}$	$\not{C}^{t}$	med	$C^{i\uparrow}$	$C^{i\uparrow}$	low
$user_{267}$	$C^{i\uparrow}$	$C^{i\uparrow}$	low	C^{t}	$C^{e\downarrow}$	hi	$\not{C}^{t}$	$\not{C}^{t}$	-	$C^{e\downarrow}$	$C^{i\uparrow}$	med/low
$user_{268}$	$C^{e\downarrow}$	C^{t}	med	$\not{C}^{t}$	$\not{C}^{t}$	-	$\not{C}^{t}$	$\not{C}^{t}$	-	$C^{e\downarrow}$	$C^{i\uparrow}$	med/low
$user_{270}$	$C^{e\downarrow}$	$C^{e\uparrow}$	med/low	C^{t}	$C^{i\downarrow}$	med	C^{t}	C^{t}	low	$\not{C}^{t}$	$C^{e\downarrow}$	med
$user_{271}$	C^{t}	C^{t}	med	$\not{C}^{t}$	$\not{C}^{t}$	-	$C^{e\downarrow}$	$C^{e\downarrow}$	med	$C^{i\uparrow}$	$C^{i\uparrow}$	low
$user_{272}$	C^{t}	C^{t}	low	$\not{C}^{t}$	$C^{i\uparrow}$	low	$C^{e\downarrow}$	C^{t}	med	C^{t}	$C^{i\uparrow}$	low

consistent. Three of these users switch in warm-start situation to *down explicit consistent*, one to *up implicit consistent*, and one to *down implicit consistent*.

Nine users stay in the same consistency level in the "*databases*" topic: two *total consistent*, three *total inconsistent*, three *up implicit consistent*, and one *down explicit consistent*. Only two users, *down explicit consistent* in the cold-start situation, change in the warm-start situation their consistency: one to *total consistent* and one to *total inconsistent*.

The last topic, "*news*", contains six users, who stay in the same consistency. Most of them, three, are *up implicit consistent*, two *total consistent*, and one *total inconsistent*. Two users change their consistency from *down explicit consistent* to *up implicit consistent*, one *up implicit consistent* user to *down explicit consistent*, one *total consistent* user to *up implicit consistent*, and one *total inconsistent* user to *down explicit consistent*.

Step 7: Interpreting the results of the consistency check. In this step, we interpret the results from the consistency check. In the cold-start phase of this study, we have no behavioral data of the users. This phase can be seen as a typical registration process of a new member to a web community. After the registration process, the user discovers the web site and gives an explicit feedback before the logout. We assume, that the user tells us her/his real interests in this explicit feedback. This data is very helpful to calculate the interest profile of the user because we have only few behavioral data from the registration process. Therefore, we will focus on the explicit feedback in the consistency check in the cold-start situation.

In the warm-start situation, the user already knows the web site and now visits web pages that are interesting to her/him. In a real world web community, this situation may not start with the second login, but in our study the number of the provided web pages is very limited. So we assume, that the user can distinguish between pages of interest and disinterest. In the warm-start situation, we will focus on the implicit feedback of a user because here we expect major changes in comparison with the cold-start situation–now the web site experience of the user will have an effect on his/her behavior (i.e., NOP).

Topic "teaching"

Measures: We observed that six users changed their consistency level between the cold-start and the warm-start situation, while five users stayed in the some level. The explicit feedback of seven users was in a higher interest category than the implicit feedback profile ($C^{i\uparrow}$, $C^{e\downarrow}$) in the cold-start situation. In the warm-start situation, we had only five users with a FP in a higher category–a reduction about 18%. Most users were in the *low* or *medium* category in the cold-start as well as in the warm-start situation.

Interpretation: Many users changed their consistency from cold-start to warm-start and most of them changed to *down explicit consistent*. This means, that the explicit feedback "corrects" the implicit feedback of the users: the users express with their explicit feedback that they are interested in this topic but after the second login (warm-start), they visited only few web pages with this topic.

Topic "research"

Measures: Five users switched their consistency between the cold-start situation and the warm-start situation, six users stayed in the same level. 45% of the users were *total consistent* and 36% were *total inconsistent* in the cold-start situation. In the warm-start situation were only two *total consistent* users and the NOP was often in a lower interest category than the FP. Most users were in the low category, but two (18%) changed from the medium into the high interest category.

Interpretation: The high percentage of *total consistent* users in the cold-start situation could mean, that the explicit feedback confirms the implicit feedback of the users.

However, the warm-start situation is different: now, only few users were *total consistent*; the FP was often in a higher category than the NOP. This means, that the users want to visit more pages with the "research" topic, expressed by the higher FP, as they did, concluded from the low NOP.

Topic "databases"

Measures: In this topic, most users stayed in their consistency, only two changed their consistency from the cold-start to the warm-start situation. The explicit feedback of six users was in a higher interest category than the implicit feedback profile ($C^{i\uparrow}$, $C^{e\downarrow}$) in the cold-start situation. In the warm-start situation only four users had an explicit feedback in a higher interest category in comparison to the implicit feedback profile. No user changed the interest category: five had low interest, three had medium interest.

Interpretation: In this topic, we observe no big differences between the cold-start and the warm-start situation. Only few users changed consistency but many were *total inconsistent* (36%); this topic does not meet the expectations of the users. Even most of the consistent users stayed in the low interest category.

Topic "news"

Measures: We observed that six users changed their consistency between the cold-start and the warm-start situation, while five stayed constant. The explicit feedback of six users was in a higher interest category than the implicit feedback profile ($C^{i\uparrow}$, $C^{e\downarrow}$) in the cold-start situation. Even in the warm-start situation, the FPs of eight users were in a higher interest category ($C^{i\uparrow}$, $C^{e\downarrow}$), this is 73% of all users. The most consistent users were located in the low interest category and only two changed the interest category.

Interpretation: The implicit feedback of the most users was in a low interest category ($C^{e\downarrow}$ and med Cat.), while the explicit feedback was located in the medium category ($C^{i\uparrow}$ and low Cat). Like the "databases" topic, the "news" topic does not meet the expectations of the users.

All Topics

We also checked the consistency of the users for all topics. Only three users ($user_{266}$, $user_{268}$, and $user_{271}$) changed to *total consistent*, which means that their behavior confirms their explicit feedback.

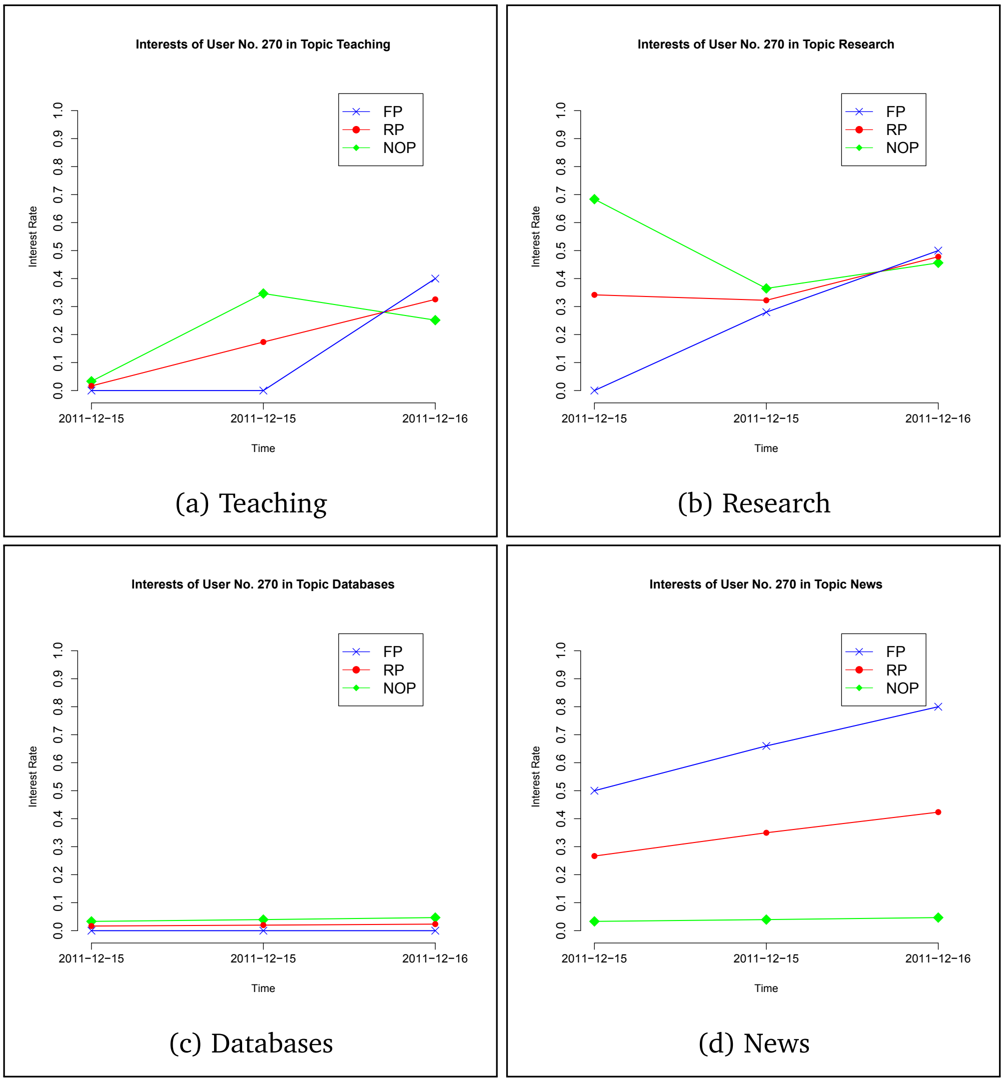
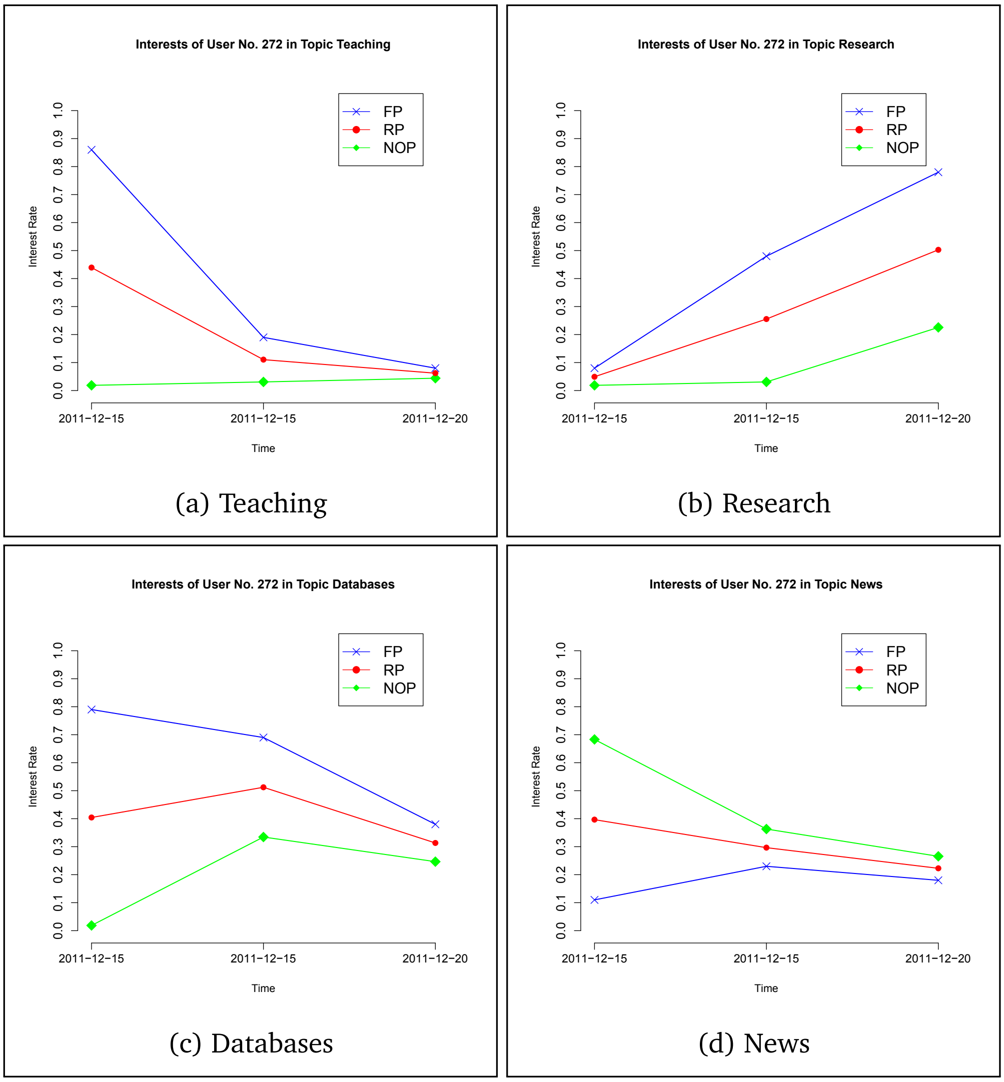


Figure 5.6: User profiles over time of user no. 270

5.3.3 User Profiles Over Time

During the cold-warm-start case study, three participants conducted the experiment supervised in a lab. Each participant performed one cold-start situation and two warm-start situations (for the detailed case study description see Paragraph 5.3.2). Figures 5.6 and 5.7 display the three interest profiles of user no. 270 and user no. 272 for the four topics “teaching”, “research”, “databases”, and “news”. The x-axis (abscissa) is labeled with the time-line, while on the y-axis (ordinate) the interest values are displayed (range: zero to one). The NOP is drawn as a green diamond, the FP as a blue “x”, and the RP as a red solid circle. Both users performed the experiment with one cold-start and



(a) Teaching (b) Research

(c) Databases (d) News

Figure 5.7: User profiles over time of user no. 272

two warm-starts, therefore, we obtained three values per profile type per topic. These three profiles are connected with lines to illustrate the changes of the interest values over time. Accordingly, the line has no meaning between the three points of time.

User$_{270}$: In the cold-start situation, the user was not interested in the "teaching" topic (Figure 5.6a), the values of all her/his interests profiles are below 0.1. In the first warm-start situation, she/he visited some web pages with the "teaching" topic, which can be seen from her/his NOP, but her/his FP shows still no interest in this topic. In the second warm-start situation, even the FP indicates interest in the topic and the values of all three user profiles are now very close together.

In the cold-start situation, the participant visited many pages with the topic "research" (NOP), but expressed with her/his FP a disinterest, see Figure 5.6b. In the warm-start situation, she/he became more and more interested in this topic (FP) and visited more web pages with this topic (NOP). Analogically to the profile-trends in the "teaching" topic, all three profiles have similar values in the second warm-start situation.

The user was completely uninterested in the "databases" topic (see Figure 5.6c). There is no visible change in all three profiles during the whole cold-warm-start case study. The values of the three interest profiles are always close together.

The FP of the user shows that she/he was interested in the "news" topic at the cold-start situation (Figure 5.6d). The interest in this topic rose with every warm-start situation. In contrast, the NOP of the user stays always very low. This means, that she/he did not visit many pages with this topic. The interest values of the profiles are very wide spread. One could interpret, that this user has high interest in "news", but the news on the DBIS web site does not meet the user's expectations.

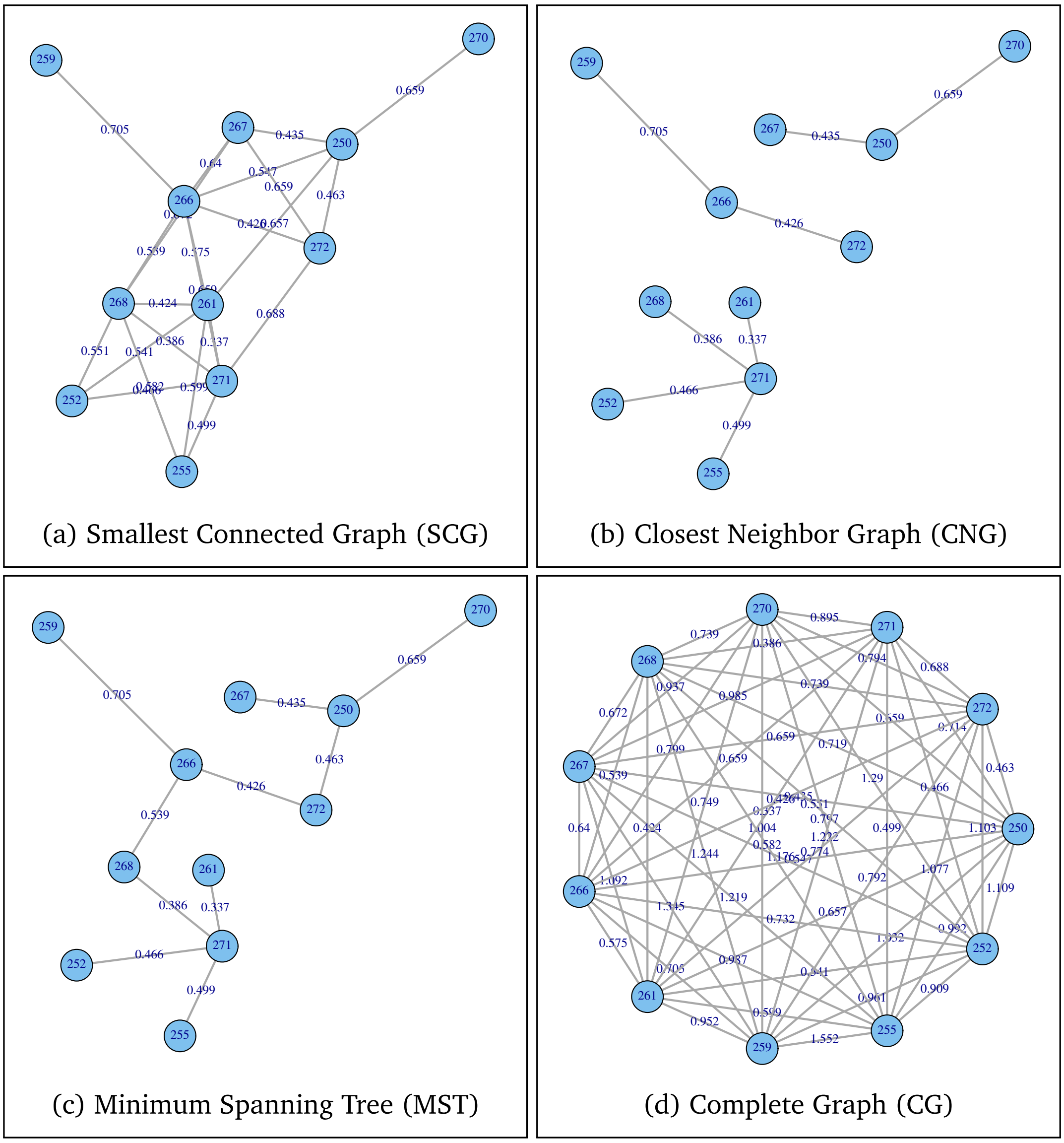
User$_{272}$: The user had high interest in the "teaching" topic (see Figure 5.7a), indicated by her/his FP, at the beginning of the case study (cold-start). The NOP shows, that the user visited only a few pages with this topic. The value of the FP drops till it reaches the value of the NOP in the second warm-start situation. At the end, the values of all profiles are nearly the same.

During the cold-start situation, all profiles of the user indicate her/his disinterest in the "research" topic, see Figure 5.7b. First, her/his FP shows a rising interest in this topic. The NOP follows this trend in the second warm-start situation.

First, in the cold-start situation, the user gave an explicit feedback (FP) that indicates high interest in the "databases" topic (Figure 5.7c). After visiting some web pages with this topic (NOP), she/he lost more and more interest, shown by her/his decreasing FP values in warm-start situation. In the second warm-start situation, her/his profiles are drawn closer together.

The pages with the "news" topic, Figure 5.7d, attracted the user in the cold-start situation–the NOP shows that she/he visited many pages with this topic. The FP is on a lower interest level but rises in both warm-start situations, while the NOP sinks. Again, the values of all interest profiles are close together in the second warm-start situation.

Observation: The most interesting fact in this cold-warm-warm experiment is that the values of the interest profiles of the users in a topic become more and more similar. In other words: the users become more and more consistent.



(a) Smallest Connected Graph (SCG)

(b) Closest Neighbor Graph (CNG)

(c) Minimum Spanning Tree (MST)

(d) Complete Graph (CG)

Figure 5.8: Similarity graphs of the cold-warm-start case study

5.3.4 User Similarity

Next, we analyzed the users with the help of similarity graphs as described in Chapter 4. The aim here is to detect important users. The importance of a single user is determined by the different algorithms that were introduced in Section 4.4.2.

First phase. In the first phase of the analysis process, we generated the similarity graphs of the users. The four graphs are displayed in Figure 5.8.

Table 5.4: Evaluation results: IDs of the users with maximum and minimum importance of every graph type (rows) for different algorithms (columns)

		Page Rank	Nearest N.D.	Dice S.C.	Jaccard S.C.	Closen. C.	Hub Score	Eigen-vector C.	Weight. Degree	Range C.
SCG	Max	266	252	261	261	266	266	266	266	266
	Min	259	270	270	270	270	270	270	259	270
CNG	Max	271	261	252,255, 261,268	252,255, 261,268	271,213	271	271	271	271
	Min	255	250,266, 271	250,266, 271	250,266, 271	259,267, 270,272	259,267, 270,272	250	259	259
MST	Max	268	261	268	252,255, 261	266	266	266	271	266
	Min	259	250	271	271	267,270	267,270	261	259	270
CG	Max	268	268	all	all	all	all	259	268	268
	Min	259	259	all	all	all	all	266	259	259

Second phase. In the second phase, we analyzed the graph, built in the first phase, with different algorithms. The aim is to find the important users in the similarity graph.

Table 5.4 displays the results of our calculations. The rows present the different graph types: *SCG* stands for *s*mallest *c*onnection *g*raph, *CNG* for *c*losest *n*eighbor *g*raph, *MST* for *m*inimum *s*panning *t*ree, and *CG* for *c*omplete *g*raph (i.e., threshold graph). For every graph type, the users with maximum and minimum importance is displayed. Every column presents one importance algorithm. We can observe the following fact in the dataset in respect to our new designed algorithms, the weighted degree and the range centrality:

In the SCG, the user no. 266 is calculated as most *important* user by the range centrality, the weighted degree, the eigenvector centrality, the hub score, the closeness centrality, and the PageRank algorithm. The Dice similarity coefficient and the Jaccard similarity coefficient calculate user no. 261 as most *important* and the nearest neighbor degree user no. 252. So, the majority of the algorithms select user no. 266 in the SCG as most *important*. The most *unimportant* user is user no. 270, calculated by all algorithms except the weighted degree and the PageRank, which computed user no. 259 as most unimportant user.

In the CNG, the user no. 271, is calculated as most *important* user by the range centrality, the weighted degree, the eigenvector centrality, the hub score, the closeness centrality, and the PageRank algorithm. The Dice similarity coefficient and the Jaccard similarity coefficient calculate user no. 252 as most *important*. The nearest neighbor degree calculates user no. 261 as *important*, which also appears in the sets of the *important* users of the Dice and the Jaccord similarity coefficient. *Unimportant* users are: user no. 259 (range centrality, weighted degree), user no. 250 (eigenvector centrality, Jaccard similarity coefficient, Dice similarity coefficient, nearest neighbor degree), and user no. 255 (PageRank).

In the MST, the range centrality, the eigenvector centrality, the hub score, and the closeness centrality select user no. 266 as most *important*. The weighted degree coefficient calculates user no. 271, the nearest neighbor degree and the Jaccard similarity coefficient user no. 261 as most *important*. The Dice similarity coefficient and the PageRank choose user no. 268 as most *important*. As *unimportant* user no. 270 is selected by the range centrality, the hub score, the closeness centrality. User no. 259 is confirmed as *unimportant* by the weighted degree and the PageRank. Also *unimportant* is user no. 271 (Jaccard similarity coefficient, Dice similarity coefficient) and user no. 250 (nearest neighbor degree).

In the CG, user no. 268 is the most *important* user for four algorithms, which select also user no. 259 as the most *unimportant* user. In contrast, the eigenvector centrality calculates user no. 259 as most *important* and user no. 266 as most *unimportant*. The Dice similarity coefficient, the Jaccard similarity coefficient, the closeness centrality, and the hub score are not able to find any *important* or *unimportant* user in the complete graph because these algorithms include not the edge weights of the similarity graph into their calculation.

5.3.5 Discussion

In the cold-warm-start case study, we compared all user profiles of the cold-start (NOP, FP, and RP), where no information about the interests of the users is available, with all user profiles of the warm-start, that includes data from more user sessions. In the feedback case study, see Section 5.2, we only compared the FPs of the users with the interest profiles calculated from a single user session. We observed the following changes in the consistencies of the users:

In all situations, *total consistent* users indicate that the web pages meet the expectations of these users. A shift to *up implicit consistent* is caused by a change of the FP into a higher ordinal category in comparison to the other interest profiles. Therefore, the topic seems to be interesting for the users but they did not perform many actions or spent not much time on web pages with this topic. That is why their NOP stays in lower ordinal category. If the users become *down implicit consistent* in the warm-start situation, the FP moved into a lower ordinal category. This could mean that the users lost interest in this topic after visiting the web site. A change to *down explicit consistent* resulted by a NOP switching to a lower ordinal category. The users performed less actions or spent not much time on web pages with this topic, which is reflected by their NOP.

Implicit consistent users have their NOP and the RP in the same ordinal category. In the cold- and warm-start situation, *up implicit consistent* users seem to have interest in the web site topic because their FP is in a higher ordinal category, but their behavior shows not the same high interest value. In the case of a change to *down explicit consistent*, the FP and the NOP switch into a lower ordinal category, which indicates that the users lost interest in the web site topic.

Explicit consistent feedback implicates that the FP and the RP are in same ordinal category. The implicit feedback, the NOP, of users that are *up explicit consistent* in both situations indicates more interest in the topic than it is expressed by their explicit feedback, the FP. In both situations, *down explicit consistent* means that the FP is in a higher ordinal category than RP and NOP. Therefore, the users express with their explicit feedback that they are more interested in the topic but they did perform only few actions on topic related pages. A change from *down explicit consistent* to *total consistent* indicates that the web pages with this topic did not meet the expectation of the users. Here, the FP dropped down into a lower ordinal category. Users that change from *down explicit consistent* to *total inconsistent* signals that the web site does not meet their expectations because their implicit feedback is different from their explicit feedback in the warm-start situation. A change from *down explicit consistent* to *up implicit consistent* implicates that the FP and the NOP switch into a higher ordinal category. This could indicate that the users become very interested in this topic during their second visit. A change from *down explicit consistent* to *up explicit consistent* means that the FP of a user switches from a higher ordinal category into a lower and the NOP changes from a lower ordinal category into a higher one.

Total inconsistent users are an indicator that the web site meets not the expectations of the users because their implicit feedback is different from their explicit feedback. A change from total inconsistent to *down explicit consistent*, FP and RP in a higher ordinal category than the NOP, indicates that the FP changed from a higher ordinal category to a lower one; the users lost interest in the topic. A change from total inconsistent to *up implicit consistent* denotes that the users perform more actions and stay longer on the web pages with this topic during their second visit. Also their FP is in a higher ordinal category, which means that they are interested in the given topic.

During the cold-warm-start case study, we observed three participants for more than two sessions. The result of this observation is that their interest profiles become more consistent over the time.

With the analysis tool for building and mining similarity graphs, we detected in the cold-warm-start case study the most important users. The two new algorithms proved to be a good alternative to the common algorithms and they showed to calculate reasonable results. With the results of the consistency check the web site owner is now able to identify weak topics and can adapt the web site accordingly. Before deploying the new web pages, she/he can test these new pages with the most important, i.e., most similar users.

5.4 Case Study: All Data of Gugubarra

In the this section, we performed a case study with all users of the DBIS web site. We analyzed the user behavior of all users during their web site visits. In contrast to the prior case studies, the web site users had not to perform any particular tasks. In this study, we wanted to test the Gugubarra Framework under real world conditions.

5.4.1 System Settings

For this study, the data of all registered visitors of the DBIS web site were analyzed. The data were collected during the period between June 2010 and July 2012. We used the same settings for the Gugubarra Framework like in Section 5.2.1 and Section 5.3.1:

a) NOP with parameter $a = b = 0.5$:
   We considered for this test, the time and the activity performed by each user equally important in the calculation of the NOP.

b) We defined four topics for the web site (see Section 3.3, step 1):
   $T_1 = teaching$, $T_2 = research$, $T_3 = databases$, and $T_4 = news$.

c) RP filter function constant with $f_l(S_k) = 1$ (see Section 3.4, step 2).
   We considered explicit user feedback with the same importance as a NOP. This is justified by the fact that we wanted to conduct a consistency check with the users that had given explicit feedback most frequently.

For each topic, zone topic weights were associated with different *zones* [HKTZ06b].

Next, we performed a framework analysis, introduced in Chapter 3, of the collected data and we searched for the most important user with the graph analytics tool introduced in Section 4.3.

5.4.2 Framework Analysis for Managing Feedback of Visitors of a Web Site

Step 1: Definition of a scope. First, we defined the scope S_k for the all users case study: the aim of the framework analysis is to understand whether the users of the web community are consistent, i.e., whether their implicit feedback correlates with their explicit feedback. For this reason, we selected the users who gave most frequently explicit feedback:

$$\begin{aligned} Cluster_{u_m} = \{&user_{63}, user_{64}, user_{75}, user_{115}, user_{127},\\ &user_{136}, user_{138}, user_{270}, user_{271}, user_{272}\} \end{aligned}$$

The cluster of topics, we like to analyze, comprised the four topics:

$$Cluster_{T_i} = \{teaching, research, databases, news\}$$

For these topics we analyzed the feedback of the users.

Step 2: Definition of a filter. We wanted the full impact of the explicit feedback on the RP calculation. Therefore, we used a constant filter function:

$$f(Cluster_{u_m}, Cluster_{T_i}) = 1$$

With this filter the explicit feedback of all users in the $Cluster_{u_m}$ has the same importance.

Step 3: Obtaining explicit user feedback. In this case study, we used all data the users of the DBIS web site collected during the period between June 2010 and July 2012. These data include the implicit feedback (behavioral information) as well as the explicit feedbacks from the registered web site visitors. There were no restrictions for the registration, everyone could register to our web site. We did not collect ethnological data, thus we cannot describe the structure of our community in detail. However, the DBIS research group is part of the University of Frankfurt. That is why we expect that most of the visitors were students, researchers, or coworkers.

Step 4: Filtering the user feedback. In this step, we apply the in step 2 defined filter function f_l on the scope S_k identified in step 1. Therefore, we calculated with Formula 3.2 the RP for all users that belong to the scope. First, we needed to calculate the NOP with Formula 2.3 per user and write the explicit feedback into the FP of each user. Second, we calculated the RP. See Section 5.2.2 for an example of the calculations.

Step 5: Clustering. Clustering the RPs of the users is only worthwhile for a large number of users. We selected only ten users for the analysis, therefore it was not necessary to cluster their RPs.

Step 6: Consistency Check. For the consistency check we used an ordinal scale with three ordinal categories:

$$\begin{aligned} D = \{&\text{“}low\ interest'', \text{“}medium\ interest'', \text{“}high\ interest''\} \text{ with} \\ &\text{“}low\ interest'' = [0.0, 0.3), \\ &\text{“}medium\ interest'' = [0.3, 0.7), \text{ and} \\ &\text{“}high\ interest'' = [0.7, 1.0] \end{aligned}$$

Next, we looked at each single user whether her/his FP or NOP and RP is located into the same category of the ordinal interest scale. We did a consistency check on the last calculated NOPs, last given FPs, and the current RPs of each user of the whole time period, as defined in Section 3.8. The results of the consistency check are presented in Table 5.5. The table displays the type of consistency, e.g., $C^{e\downarrow}$, of the users and the ordinal category in brackets ($med = medium$) where the consistent user profiles are located.

Figures A.9 to A.12 in Appendix A, display the RP as red solid circle, the NOP as green diamond, and the FP as a blue plus, for each user per topic. The x-axis displays both the interest value of the users in metric form and the ordinal categories. On the y-axis the user IDs are depicted.

When we applied the consistency check for the “*teaching*” topic we obtained the results displayed in Figure A.9 (shown in Appendix A). The results are summarized in Table 5.5, where we can see that five users (equals 50% of all users) are *total consistent* (C^t), three users are *down explicit consistent* ($C^{e\downarrow}$), and one user is *total inconsistent* ($\not{C}^t$). Most of the users are located in the *medium*, three in the *low*, and one in the *high* interest category.

In the results (Figure A.10) of the consistency check for the “*research*” topic, we obtained the results presented in Table 5.5. The table shows that in the “research” topic six of the users (60%) are *total inconsistent* and only one (10%) is *total consistent* in the

Table 5.5: Result of the consistency check for the all users case study

	Topic			
user$_{\text{id}}$	**Teaching**	**Research**	**Databases**	**News**
$user_{63}$	C^t (low interest)	$\not\subset^t$	$C^{e\downarrow}$ (med interest)	$C^{i\uparrow}$ (low interest)
$user_{64}$	$\not\subset^t$	$C^{i\uparrow}$ (low interest)	C^t (low interest)	C^t (low interest)
$user_{75}$	C^t (med interest)	$\not\subset^t$	$C^{i\uparrow}$ (low interest)	$C^{i\uparrow}$ (low interest)
$user_{115}$	$C^{e\downarrow}$ (high interest)	$\not\subset^t$	$C^{e\downarrow}$ (high interest)	$\not\subset^t$
$user_{127}$	$C^{e\downarrow}$ (med interest)	$C^{e\downarrow}$ (med interest)	$\not\subset^t$	$C^{e\downarrow}$ (med interest)
$user_{136}$	$C^{e\downarrow}$ (med interest)	$\not\subset^t$	$C^{e\downarrow}$ (med interest)	C^t (med interest)
$user_{138}$	C^t (low interest)	$\not\subset^t$	C^t (low interest)	$\not\subset^t$
$user_{270}$	C^t (low interest)	$C^{i\downarrow}$ (med interest)	C^t (low interest)	$C^{i\uparrow}$ (low interest)
$user_{271}$	C^t (med interest)	$\not\subset^t$	$C^{e\downarrow}$ (med interest)	C^t (low interest)
$user_{272}$	$\not\subset^t$	C^t (low interest)	$\not\subset^t$	$C^{i\uparrow}$ (med interest)

low interest category. Two users are *implicit consistent*, one *up* with *low* interest and one *down* with *medium* interest. In the *medium* interest category only one *down explicit consistent* user is located.

When the consistency check was applied for the "*databases*" topic (see Figure A.11 in Appendix A), the following results were obtained (Table 5.5). In this topic, we have three *total consistent* users (30%) and two *total inconsistent* users. *Down explicit consistent* are four users, *up implicit consistent* is only one user. The distribution of the interest categories is as follows: four times *low interest*, three times *medium interest*, and once *high interest*.

The results of the consistency check for the "*news*" topic (see Figure A.12 in Appendix A) are displayed in Table 5.5. We observe three *total consistent* users (30%) and two *total inconsistent* users. Four users are *up implicit consistent* and one is *down explicit consistent*. Most of the consistent users are in the *low interest* category (50%), the rest (30%) is located in the *medium interest* category.

Step 7: Interpreting the results of the consistency check. In this step, we interpret the results of the consistency check. In this all users case study, we are interested in the current state of our web community. Therefore, we analyze only the current NOP, FP, and RP of a user.

Topic “teaching”

Measures: Many *total consistent* users only had *low interest* in the “teaching” topic but there were also many users with the FP in a higher interest category than the NOP.

Interpretation: Cause of the high percentage of *total consistent* users, the web page with the “teaching” topic seems to meet the expectations of the users. These users were not very interested but they found what they want to find. The web site owner could do some minor changes in order to make the web site more interesting for the users with a FP in a higher interest category.

Topic “research”

Measures: In this topic, the majority of the users was *total inconsistent*.

Interpretation: From their explicit feedback, the users seem to be interested in the “research” topic. But the NOP with the implicit feedback was mostly in the low interest category. The web site owner should consider to reconstruct this section of the web site because the visitors seems to be very unpleased about the pages.

Topic “databases”

Measures: The “database” topic had many *down explicit consistent* users and *total consistent* users.

Interpretation: The NOPs of the *down explicit consistent* users were more often in a lower interest category than the other interest profiles. The web site owner should provide more pages with the content “databases” to encourage the users to visit more pages with this topic.

Topic “news”

Measures: In the “news” topic, the FP of many users was in a higher interest category than their NOP. Many users were in the low interest category.

Interpretation: For their FP, the users are interested in this topic. Therefore, the web site owner should provide more news items to them.

All Topics

We checked the consistency of the users for all topics too. $User_{115}$ had high interest in two topics and her/his profiles were very consistent. Only the NOP was mostly located in a lower interest category. $User_{138}$ was two times *total consistent* in the low interest category and two times *total inconsistent*. $User_{270}$ was the only consistent user in any topic.

5.4.3 User Similarity

Next, we analyzed the users with the help of similarity graphs as described in Chapter 4. The aim here is to detect important users. The importance of a single user is determined by the different algorithms that were introduced in Section 4.4.2.

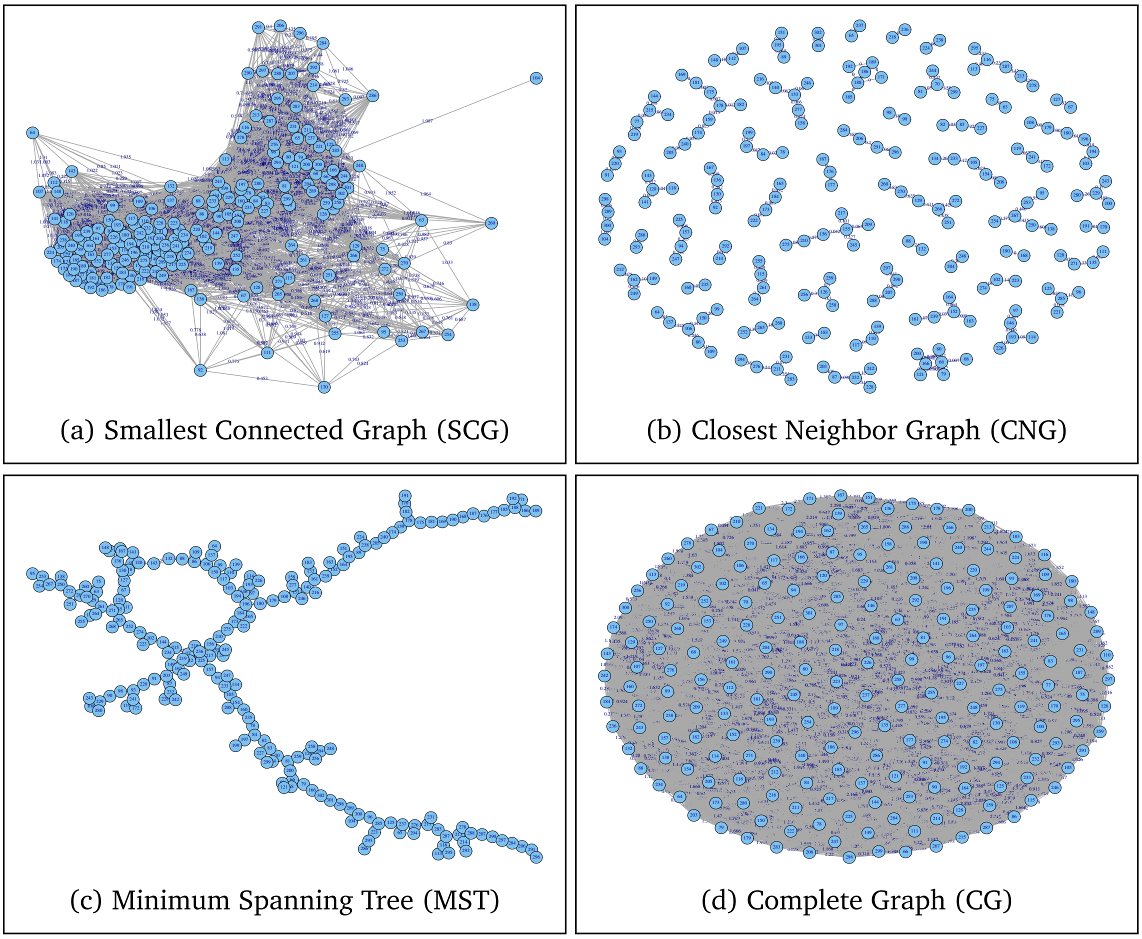
First phase. In the first phase of the analysis process, we generated the similarity graphs of the users. The four graphs are displayed in Figure 5.9.

Second phase. In the second phase, we analyzed the graph, generated in the first phase, with different algorithms. The aim here is to detect the important users in the similarity graph.

Table 5.6 displays the results of our calculations. The rows present the different graph types: *SCG* stands for *s*mallest *c*onnection *g*raph, *CNG* for *c*losest *n*eighbor *g*raph, *MST* for *m*inimum *s*panning *t*ree, and *CG* for *c*omplete *g*raph (i.e., threshold graph). For every graph type, the user/users with maximum and minimum importance is displayed. Every column presents one importance algorithm. We can observe the following fact in the dataset in respect to our new designed algorithms, the weighted degree and the rang centrality:

In the SCG, the range centrality calculates user no. 220 as most *important* user. The weighted degree, the closeness centrality, and the PageRank select user no. 93 as most *important*. User no. 223 is *important* for the eigenvector centrality and the Dice similarity coefficient. The hub score chooses user no. 91 and the nearest neighbor degree user no. 216 as most *important*. The majority of algorithms calculate the same *unimportant* user (user no. 104), only the nearest neighbor degree centrality differs (user no. 138).

In the CNG, the range centrality and the closeness centrality calculates the same *important* user (user no. 178). The same *unimportant* users (user no. 63 and user no. 75) are selected by the range centrality, the hub score, the closeness centrality, the Jaccard and the Dice similarity coefficient, and the nearest neighbor degree. The results of the



(a) Smallest Connected Graph (SCG)

(b) Closest Neighbor Graph (CNG)

(c) Minimum Spanning Tree (MST)

(d) Complete Graph (CG)

Figure 5.9: Similarity graphs of all Gugubarra users

weighted degree for the most *important* user is no. 66 and for the most *unimportant* user no. 104.

In the MST, the results of the range centrality equals the closeness centrality, while the weighted degree calculates the same *unimportant* user as the PageRank. The range centrality, the hub score, and the closeness centrality select user no. 296 as most *unimportant* one.

In the CG, user no. 241 is computed as the most *important* user by all algorithms, except for the eigenvector centrality. User no. 104 is the most *unimportant* user for the rang centrality, the PageRank, and the nearest neighbor degree. The eigenvector centrality calculates the opposite results. The Dice similarity coefficient, the Jaccard similarity coefficient, the closeness centrality, and the hub score are not able to find an *un-/important* user in the complete graph because these algorithms do not include the edge weights into their calculation.

Table 5.6: Evaluation results: IDs of the users with maximum and minimum importance of every graph type (rows) for different algorithms (columns)

		Page Rank	Nearest N.D.	Dice S.C.	Jaccard S.C.	Closen. C.	Hub Score	Eigen-vector C.	Weight. Degree	Range C.
SCG	Max	93	216	223	232	93	91	223	93	220
	Min	104	138	104	104	104	104	104	104	104
CNG	Max	155	169	79,80, 121,200	79,80, 121,200	178	66	300	66	178
	Min	68	63,65, 67,...	63,65, 67,...	63,65, 67,...	63,65, 67,...	63,65, 67,...	270	104	63,75
MST	Max	241	169	68,80, 121,200	68,80, 121	225	261	129	66	225
	Min	104	293	112,229, 232	112,229, 232	296	296	300	104	296
CG	Max	241	241	all	all	all	all	104	241	241
	Min	104	104	all	all	all	all	241	79	104

5.4.4 Discussion

To test the Gugubarra Framework, we performed a consistency check with the data of all users of our web site. One result of the cold-warm-start case study, Section 5.3, was that the user profiles became more and more consistent over the time. In the all users case study, we did not compare the user profiles of different sessions. Instead, we performed a consistency check on the current state of the web community (snapshot). For the consistency check we chose ten users that gave the most frequent explicit feedback. We observed the following consistencies among the users of the on-line community:

Total consistent users have all their interest profiles in the same ordinal category. It is very likely that they will not change their consistency level, as we discovered in Section 5.3.3. Therefore, their ordinal category reflects their interest level in a topic. We observed no total consistent users in the high ordinal category.

The FP and RP of users that are *explicit consistent*, are in the same ordinal category. We observed only *down explicit consistent* users among our community members. These users claim to have more interest in a topic (FP) than we have derived from their behavioral data (NOP). In contrast, *up explicit consistent* users have a high interest value, calculated from their behavioral data, and a low interest value from their explicit feedback.

Without comparing interest profiles like in the prior case studies, the interpretation of the consistency level "*implicit consistent*" is very similar to the consistency level "explicit consistent": the explicit feedback of *up implicit* and *down explicit consistent* users indicates that they are more interested in a topic than we calculated from their implicit feedback (behavioral data). *Down implicit* and *up explicit consistent* users seem to be more interested in a topic from their implicit feedback than they express by their explicit feedback.

Total inconsistent users have a completely different implicit feedback in comparison to their explicit feedback. The behavior of these users is very difficult to estimate and it will be hard for the web site owner to provide interesting content for these users. With the right content, their consistency level "total inconsistent" would change to another, more consistent, level.

With the analysis tool for building and mining similarity graphs, we detected in the all users case study the most important and the most unimportant users. In contrast to the consistency check, we used for the calculation of the most un-/important users the data of all web site visitors. The two new algorithms proved to be a good alternative to the common algorithms. Moreover, they calculated reasonable results.

5.5 Conclusion and Future Work

Conclusion. In this chapter, we conducted three case studies to proof the applicability of the new extensions of the Gugubarra Framework: the framework analysis for managing feedback of visitors of a web site (see Chapter 3) and the tool for building and mining similarity graphs (see Chapter 4).

The first case study focused on the *explicit user feedback*. We compared two explicit feedbacks of users during their first web site experience. The first feedback was given before the user visited the web site the first time (*cold-start situation*), the second afterwards. With the second study, the *cold-warm-start* case study, we compared all user profiles (NOP, FP, and RP) of the cold-start, where no information about the users interests is available, with all user profiles of the warm-start, which includes data from previous user sessions. In the last case study, we performed a consistency check with the data set of all users of our web site. We did not compare the user profiles of different sessions. Instead, we performed a consistency check on the current state of the web community (snapshot). In brief, the first and the second case study observed the changes in the feedbacks of a user, the third focused on the current state of the user interest profiles. In all three case studies, we calculated the most important and the most unimportant users of the web community too. We draw the following conclusion:

Consistency check. From this set of studies, we observed different consistency levels beyond the web site users. These consistency levels might be interpreted in the following way:

Structural problem: under the term of "structural problem" we summarize all problems like a bad menu structure of a web site, an unclear layout of a web site, or an illogical order of the single web pages. An indicator of such a structural problem is, when the FP of a user is in a higher ordinal category than the other interest profiles. This means in detail, that a user claims to be more interested in a topic by her/his explicit feedback, than the Gugubarra Framework calculates from her/his implicit feedback. The following consistencies describe this phenomenon:

- A change from total consistent (C^t) to down explicit consistent ($C^{e\downarrow}$)
- A change from total inconsistent ($\not{C}^t$) to down explicit consistent ($C^{e\downarrow}$)
- Up implicit consistent ($C^{i\uparrow}$)
- Down explicit consistent ($C^{e\downarrow}$)

Topic mismatch: the user understands a topic in a different way than the web site owner who defines the topics of the zones. Therefore, the topic term does not match to the content of the web pages from the view of the web site user. As a consequence, it is not possible to compare the implicit feedback, which reflects the topic terminology of the web site owner, with the explicit feedback, which reflects the topic terminology of the web site user. This could be the case, when the NOP of a user is in a higher ordinal category in comparison with the other user profiles. In detail, the user seems to be interested in a topic, calculated from her/his behavioral data (NOP), but in her/his explicit feedback (FP) she/he claims to have less interest in this topic. The following consistencies describe this phenomenon:

- A change from total inconsistent ($\not{C}^t$) to up explicit consistent ($C^{e\uparrow}$)
- A change from up implicit consistent ($C^{i\uparrow}$) to total consistent (C^t)
- A change from down explicit consistent ($C^{e\downarrow}$) to total inconsistent ($\not{C}^t$)
- Up explicit consistent ($C^{e\uparrow}$)
- Down implicit consistent ($C^{i\downarrow}$)

Lost interest: after discovering the web site, the user loses interest in a topic. An indication for this could be a change in the explicit user feedback, moreover, when the explicit feedback changes to a lower ordinal category. This is the case in the following consistency levels:

- A change from total consistent (C^t) to down implicit consistent ($C^{i\downarrow}$)
- A change from up implicit consistent ($C^{i\uparrow}$) to down explicit consistent ($C^{e\downarrow}$)
- A change from down explicit consistent ($C^{e\downarrow}$) to up explicit consistent ($C^{e\uparrow}$)
- A change from total inconsistent ($\not{C}^t$) to down explicit consistent ($C^{e\downarrow}$)
- Total inconsistent ($\not{C}^t$)

Stable interest: the user seems to be generally interested in a topic and the interest does not change over time. This is the case in the following consistency levels:

- A change from down explicit consistent ($C^{e\downarrow}$) to total consistent (C^t)
- A change from down explicit consistent ($C^{e\downarrow}$) to up implicit consistent ($C^{i\uparrow}$)
- A change from up explicit consistent ($C^{e\uparrow}$) to total consistent (C^t)
- A change from down implicit consistent ($C^{i\downarrow}$) to total consistent (C^t)
- A change from total consistent (C^t) to up implicit consistent ($C^{i\uparrow}$)
- A change from total inconsistent ($\not{C}^t$) to up implicit consistent ($C^{i\uparrow}$)
- Total consistent (C^t)–the ordinal category reflects the interest level of the user

In summary, with the consistency check we are able to analyze the users of the web community. With its results the web site owner can detect weak and strong points of her/his web portal.

User similarity. Since there is no objective measurement for importance, we evaluated our two new algorithms, the weighted degree and the range centrality, by comparing the results with previously published algorithms. Within the three case studies, conducted in this chapter, we calculated in each study the most important and most unimportant users of the web community. The web community of the first and the second case study consisted of ten participants respectively. The last case study took into account the entire on-line community of the DBIS web site, more than 200 users.

However, every algorithm calculates the importance in a different way. Most of the algorithms are not designed for similarity or even weighted graphs. Thus, evaluation and comparison of the single results is challenging.

By definition of the weighted degree, the most important user is the vertex with the most connections. If there are vertices with the same number of connections it takes the vertex with the lowest average edge weight. In the feedback and the cold-warm-start case study, the results of all algorithms are very similar. This can be explained by the small number of users. The results in the last case study with its large web community are different: the PageRank and the hub score calculate similar users as most important. In contrast to the hub score, the weighted degree algorithm can calculate the most important users in a complete graph (all users have the same number of connections) because it considers the edge weights of the connections.

By definition of the range centrality, the most important user has many connections in comparison to the other users of the graph, short distances to her/his neighbors, and low edge weights. In the feedback and the cold-warm-start case study, the results of all algorithms are very similar. In the third case study, the results are partly comparable with the PageRank and the closeness centrality. In contrast to the closeness centrality, the range centrality algorithm can calculate the most important and most unimportant users in a complete graph too.

In summary, we think that our two new algorithms are a good alternative for mining similarity graphs. Especially in comparison to the Dice similarity coefficient, the Jaccard similarity coefficient, the closeness centrality, and the hub score both algorithms can calculate the most important and the most unimportant users in a complete graph.

Future Work. In future, we want to conduct more case studies: on the one hand, to detect more indicators for weak/strong points of a web site. On the other hand, to reduce the complexity of the consistency check. During the three case studies it became obvious that some consistency levels have the same expressiveness, so that the set of consistencies can be reduced. This would make it easier for the web site owner to conduct a consistency check.

Furthermore, we plan a case study that focuses only on the similarity of the users in a on-line community. With such a study, we want to define an objective measurement for importance. So, it would be possible to compare the results of each algorithm of our tool for building and mining similarity graphs.

Table 5.7: Contribution summary

Subject	***Detail***
Case Studies	We demonstrated the applicability of the framework analysis for managing the feedback of web site visitors and the tool for building and mining similarity graphs with the help of three case studies: – First case study focused on the explicit user feedback in a cold-start situation (Section 5.2). – Second case study compared a cold-start situation with a warm-start situation (Section 5.3). – Third case study used the complete data set of the users of the DBIS web site (Section 5.4).
Consistency Check	We discovered with the help of the consistency check indicators for (Section 5.5): – *Structural problem:* the web site has a bad structured menu, unclear layout, or illogical order of the single pages. – *Topic mismatch:* the web user understands a topic in a different way than the web site owner does. – *Lost interest:* after discovering the web pages with a topic, the user lost interest in this topic. – *Stable interest:* the user seems to be interested in a topic and the interest does not change over time.
User Similarity	We detected the most important and most unimportant users of the web community in all case studies. The new algorithms are a good alternative in comparison to the common algorithms for mining similarity graphs. (Section 5.2.3, 5.3.4, 5.4.3)

6 Mouse-Tracking

To gain more behavioral data from the web site visitor, the Gugubarra Framework is extended with the possibility to track the mouse activities of the web users. First, the potential of the mouse-tracking technology in respect to the Gugubarra Framework is discussed. Second, the practical realization and implementation is presented. This chapter is based on the publication of Schefels et al. [SES12].

6.1 Introduction

We present a method to extract implicit data of registered users of a web site with the help of mouse-tracking. This allows us to generate more accurate interest profiles of visitors of a web site and to obtain a solid basis for the calculation of user interests or trend detection in the Web. With this method, web site owners have the opportunity to adjust their sites to the interests of their users. Additionally, they can detect trending topics and extend the content for these topics on their web pages accordingly.

In this research, we use the Gugubarra Framework, introduced in Chapter 2 and in [MWTZ04, HZ08], a web analytics system, developed by DBIS[1] at the Goethe-University Frankfurt. The goal of the system is to actively help the owner/manager of a web site to better understand the interests of users registered on her/his web site. The

[1] `http://www.dbis.cs.uni-frankfurt.de/`

previous version of the system analyzed extended server log files for the computation of implicit interest profiles. In this chapter, we extend the Relevance Profile (RP), first introduced in Chapter 3 and in [SZ10, SZ12], with data generated by mouse activities of a web site user. The RP is a container for all available interest profiles of a user. The mouse-tracking enhances several concepts of Gugubarra, including zones, actions, and duration, which are explained in Chapter 2 and in the publication of Hoebel et al. [HMS+09].

Mouse-tracking has several advantages over eye tracking [CAS01]: First, it is less prone to the Hawthorne-effect[2] [FK78]. Second, it does not need complex calibration and it is less error-prone (in acquiring positional data) than eye tracking. Third, it is more precise and eases the mapping to semantic zones.

In what follows, we assume that users are aware and have granted permission that implicit data is collected and kept in their profile for them.

The rest of this chapter is structured as follows: Section 6.2 presents related work. Section 6.3 recalls the basic concepts of Gugubarra that will be used in the rest of the chapter. Section 6.4 and Section 6.5 describe the main contribution of this chapter, the integration and implementation of mouse-tracking in the Gugubarra Framework. The next section, Section 6.6, shows an evaluation of Gugubarra mouse-tracking module and Section 6.7 presents the conclusion and outlines future work.

6.2 Related Work

Most mouse-tracking systems in research are limited to the collection and evaluation of raw mouse data. Our approach is different: we use the Gugubarra Framework and we can, therefore, enrich the pure mouse-tracking data with additional information about a web page.

Cox and Silva conducted two experiments in [CS06] and found out that users often use the mouse cursor to tag potential targets while searching through a menu structure. In our system, we interpret this tagging as indication that the user is interested in a menu item.

Arroyo et al. present in [ASW06] the mouse-tracking system “MouseTrack”, which includes a configuration and visualization tool. In contrast to our approach, this system is proxy based, which means that a user has to modify her/his browser configuration to get tracked. Atterer and Schmidt show in [AS07] with their mouse-tracking system [AWS06] that the willingness of users to participate in a usability test depends

[2]Hawthorne-Effect: the knowledge of a subject being watched changes its behavior.

much on the used logging technology: if the participants have to install additional software or have to change their browser configuration, many are not willing to attend the test. Atterer and Schmidt who use a proxy based mouse-tracking system, concluded that the easiest way to recruit study participants would be to develop a tracking system that needs no configuration on the client side. Our system is directly integrated into the content management system Joomla! so that no modification on the client side is necessary.

In [TH07], Torres and Hernando developed a mouse-tracking system named “smt”. To track the mouse usage, the web site owner has to include a JavaScript code manually in every monitored web page. The “smt” system is similar to our work, except that Torres and Hernando integrated their system not into a content management system. Therefore, in our case the JavaScript code of the mouse-tracking system is automatically included in every web page managed by Joomla!. Additionally, we use the data of the mouse-tracking system to build user profiles that Torres and Hernando suggest in their future work section.

Rodden et al. analyzed in [RFAS08] the pattern of coordination between users’ eye movements and mouse movements when scanning a web search result page. Guo and Agichtein [GA10] extended this work to predict gaze position from mouse movements. Both publications show a correlation between eye and mouse movements. For this reason, mouse-tracking is a good alternative for expensive eye tracking, which is practicable only in labs. It is also a good additional source for implicit user data for profiling.

To understand the idea of the Gugubarra Framework, the next section will call up its main concepts. For a more detailed description see Chapter 2.

6.3 Gugubarra and User Profiles

In this section, we briefly recall the main concepts of the Gugubarra Framework, which will be used throughout the rest of this chapter. Gugubarra is a prototype system developed by DBIS at the Goethe-University Frankfurt, with the goal to actively help the owner/manager of a web site in better understanding the supposed interests of users registered on her/his web site.

For each registered user Gugubarra generates two interest profiles, which collect data related to the user:

- A *Feedback Profile* (FP), which stores the data explicitly given by a user. For that, we ask the users from time to time about their interests in respect to a set of predefined topics, see Chapter 2, Section 2.4.1.

- A *Non-Obvious Profile* (NOP) that stores behavioral data not explicitly given by the user, but automatically created by analyzing the user behavior on the web site. The behavioral data stored in the NOP indicates, for example, what pages a user has visited, and what actions she/he has performed on that web page. Most of this information is extracted out of the web server log, but Gugubarra has refined the common click-stream analysis [WH06, JHW07], by extending it with new concepts, namely: *zones*, *topics*, *actions*, and *weights*. These concepts are explained in detail in Chapter 2, Section 2.3. The calculation of a *NOP* is shown in Chapter 2, Section 2.4.2.

- In Chapter 3, we introduced the *Relevance Profile* (RP). The RP is calculated by integrating the NOP and the FP of a user. Furthermore, the FP is filtered by the filter function f_i to the relevant scope S_k. The benefit of the RP is that it flexibly integrates both, calculated data as well as feedback of the user into one single user profile. The calculation of an RP is shown in Chapter 3, Section 3.6.

6.4 Mouse-Tracking in the Context of Gugubarra

The method that we propose in this chapter, includes implicit data of the mouse activities of a user into her/his RP profile. Therefore, we define a basic mouse-event e_i as a member of all possible mouse-events E as:

Definition 6.1 *Mouse-event e_i:*

$$e_i := \langle Typ_i, Timestamp_i, Attributes_i \rangle \in E.$$

with Typ_i the type of the event, e.g., a mouse click, $Timestamp_i$ point in time of the event, and $Attributes_i$ a set of associated attributes, e.g., screen coordinates and identifier of the pressed button.

A mouse trail T_i is defined as a sequence of (basic) mouse events $e_{i,j}$:

Definition 6.2 *Mouse trail T_i:*

$$T_i := [e_{i,j} : j = 1..n] \; with \; t(e_{i,j}) < t(e_{i,j+1}).$$

A mouse trail T_i starts with $t(e_{i,1})$, the loading of a page, and ends with $t(e_{i,n})$, the unloading of the same page.

Furthermore, we define the duration of a page visit as:

Definition 6.3 *Durations for page visits:*

$$\delta_{i_f}(\delta_i) := \delta_i - \sum_{\delta_j \in \Delta_i} \delta_j$$

Where Δ_i is the set of detected (assured) inactivities during any duration interval δ_i. For page visits, the visit time can be determined more accurately by mouse-tracking than by a click-stream when using the load and unload events of the web page. The implementation of these considerations follows in the next section.

In contrast to pure click-stream analysis, with our methods it is now possible to extend the set of actions by *lingering* [ML01], which is a sequence of events [enter,...,leave] on an interactive element. Furthermore, it is possible to detect, whether the mouse button was pressed but released outside an element. Such lingering is considered a hesitation and since hesitation in general is a sign/result of complex neuronal evaluation, it is interpreted as an indicator for interest. It is assumed that the longer a user lingers, the greater the interest in a topic and thus the action's weight should be scaled by the duration of *lingering*.

With mouse-tracking, we detect zone visits, which will start with the first entry of *any* HTML element belonging to the zone. It naturally ends when leaving the last element that is a member of the zone. At the same time, this is a first entry of any element that is not part of that same zone.

Another opportunity of mouse-tracking is improving all duration measurements for (page) visits, lingering, and page views. It is further possible to eliminate times of assured inactivity (as opposed to lingering). Such inactivities take place, when the user minimizes the browser window or switches to another program. By detecting resize, blur, and focus events on the browser's window object, periods of assured inactivity can be identified and eliminated from determined durations δ_i for visits.

6.5 Implementation

The Gugubarra Framework consists of two parts: the Gugubarra Designer and the Gugubarra Analyzer, as described in Chapter 2 and in Hoebel's PhD thesis [Hoe11].

The first part, the Gugubarra Designer, is implemented as part of the Joomla! content management system (CMS). Besides using the common functions of this CMS, here the web site owner can add topics, zones, and actions to her/his web site. These additional meta data are stored in XML-files on the server.

The second part, the Gugubarra Analyzer, is a separately operating web client. It downloads log files as well as the meta data files from the Gugubarra Designer and calculates user profiles. It is a web application so that the results of the profile calculation can be visualized in a web browser.

The next subsections describe how we extend the Gugubarra Framework with the mouse-tracking functionality.

6.5.1 Gugubarra Designer

As noted earlier (see Section 2.5.1), the Gugubarra Framework is directly integrated into the site's CMS. The current prototype uses Joomla! as CMS and the Gugubarra Designer are built as an extension to Joomla!.

One of the major questions regarding mouse-tracking is the acquisition of the actual tracking data. The most obvious alternatives seem to be either a (native) browser plug-in or a solution based on JavaScript.

While a native browser plug-in is certainly the most flexible solution, since it gives access to most internal data structures and the browser's internal event system, possibly even to the GUI's event queue, this approach comes with some major drawbacks: it usually requires the user's help to get the plug-in installed. At the same time, the user has no means to restrict the functions of a plug-in to a specific web site. Additionally, if a plug-in needs a rights escalation for some specific functionality, the user needs to be able to provide these extended rights to the plug-in. If the user lacks the necessary rights (or capabilities) the required functionality will not be available. Peeking into the GUI's system event queue is an example for a functionality that typically needs a rights escalation.

The most severe drawback though is the complete lack of portability and platform independence as well as the need to develop custom plug-ins for each browser. Only few browsers support cross-platform plug-ins. However, these plug-ins are usually very limited in the available API and run in a sandbox, which would render their advantages mostly useless.

In contrast, a JavaScript-based approach is very portable. Since JavaScript is standardized as well as DOM Events, etc., a JavaScript based solution can be built on established and accepted standards. Even though some events might not be readily available or not (yet) standardized it gives a solid basis for an implementation while ongoing efforts in the evolution of standards will straighten out current issues. As an example,

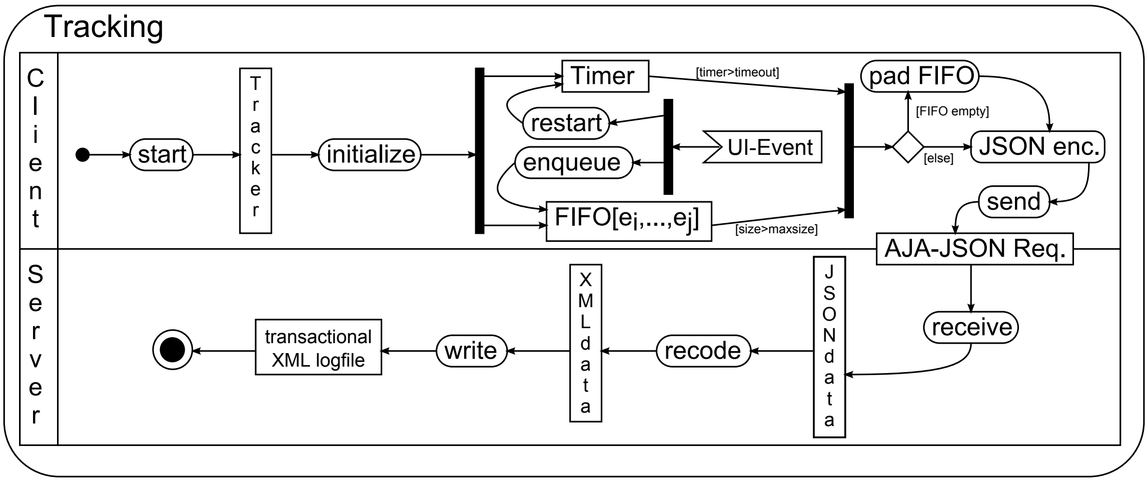


Figure 6.1: Activity diagram of the mouse-tracking implementation

currently the standards do not yet specify how certain events should be handled when using tabbed browsing and tabs are switched.

Furthermore, this approach is less intrusive on the user's side, which mostly eliminates unwanted behavioral changes by the user. Especially as Hawthorne [FK78] shows, the knowledge of a subject being watched changes its behavior.

On the downside, the abilities of JavaScript are strongly limited by the available APIs (like DOM Events) and the raw execution speed is clearly slower than with a native plug-in. The latter problem can be mitigated with technologies like JIT compilation[3], which are becoming more common with most browsers lately.

As some tasks in JavaScript are rather cumbersome and there are still certain differences between browsers (esp. earlier versions) it is common to use a JavaScript framework. Since such frameworks often extend functionality, multiple frameworks might offer identical (extended) syntax with different semantics, which could yield conflicting or unspecified behavior. As Joomla! is built around mootools[4] and favors its use, it became our choice as well.

Our solution, as shown in Figure 6.1, uses a small FIFO queue in the browser client, which stores the mouse events, defined in Definition 6.1, as they occur. When the queue stores a certain number of events or if there have not been new events in a certain timespan, the local queue is sent as an AJAX-like request and then flushed. We use

[3]JIT: just in time compilation transforms a scripting language into a native optimized code, instead of interpreting it.

[4]`http://mootools.net/`

JSON[5] as data format (see Figure 6.2), as it is more compact than XML and suits our needs perfectly. The idea of a local queue is based on the fact that an HTTP request possibly involves opening a new connection (depends on pipelining) and that we want to reduce the overhead introduced by the request's headers. Additionally, the events are filtered in the client to only include relevant data, to minimize the amount of data being sent and because most browser's native JSON stringifiers can not cope with object methods (which exist in native JS DOM Event Objects). If an empty FIFO is sent, we pad it with a special keep-alive event. In case the client crashes or something similar happens, this helps us in determining the last point in time, when there was still activity in the client.

On the server side, the JSON encoded event list is then transformed into an XML notation, as seen in Figure 6.2, associated with the proper session of the user and stored into a transaction-based log for further processing by the Gugubarra Analyzer.

6.5.2 Gugubarra Analyzer

The Gugubarra Analyzer provides a web application to calculate and manage the user profiles. These profiles are based on the information in the XML-files created by the Gugubarra Designer as described before.

For each user session, one mouse trail is saved in an XML-file (see Definition 6.2). Each trail consists of a number of mouse events and their meta data, including timestamp, name of the HTML element, etc. However, we can not simply use these meta data to create more specific user profiles: on the one hand, the design of Gugubarra causes some limitations, on the other hand there are some technical constraints. Below, we show some examples of these limitations and our solutions.

One technical limitation comes with the information '*timestamp*' that provides the Unix-time of every saved event. The source from which these timestamps are retrieved varies. As the beginning and the end of each session are located on the web portal, the events *sessionstart* and *sessionend* get timestamped from the server's clock. All other events are collected on the client-side, so their timestamp originates from the client's clock. An equation about the timestamps of these client sided collected events is possible without any restrictions. Assuming that not all clocks are perfectly synchronized, it is not advisable to compare timestamps from different sources. Consistent with this problem, we just compare *sessionstart* and *sessionend*, which both get timestamped from the server, to get the duration of the whole session.

According to the Gugubarra concept of zones, we want to register a zone visit includ-

[5] JavaScript object notation - RFC 4627 and ECMAScript Standard

JSON-Notation:

```
[{ "type" : "mouseover",
   "timestamp" : 1317326640.51,
   "x" : 509,
   "y" : 384,
   "tagName : "DIV",
   "tagID" : 102,
   "tagClass" : "guguZone"
},
{ "type" : "mousemove",
  "timestamp" : 1317326640.511,
  "x" : 509,
  "y" : 384,
  "tagName : "DIV",
  "tagID" : 102,
  "tagClass" : "guguZone"
},
{ "type" : "mouseout",
  "timestamp" : 1317326640.513,
  "x" : 479,
  "y" : 406,
  "tagName : "DIV",
  "tagID" : 102,
  "tagClass" : "guguZone"
}]
```

XML-Notation:

```
<Event type="mouseover">
  <timestamp>1317326640.51</timestamp>
  <x>509</x>
  <y>384</y>
  <tagName>DIV</tagName>
  <tagID>102</tagID>
  <tagClass>guguZone</tagClass>
</Event>
<Event type="mousemove">
  <timestamp>1317326640.511</timestamp>
  <x>509</x>
  <y>384</y>
  <tagName>DIV</tagName>
  <tagID>102</tagID>
  <tagClass>guguZone</tagClass>
</Event>
<Event type="mouseout">
  <timestamp>1317326640.513</timestamp>
  <x>479</x>
  <y>406</y>
  <tagName>DIV</tagName>
  <tagID>102</tagID>
  <tagClass>guguZone</tagClass>
</Event>
```

Figure 6.2: Side by side comparison of JSON and XML notations

ing its duration. The Gugubarra Designer stores two sets of meta data of a complete zone visit. The first one is the x- and y-coordinates acquired by mootools. However, these coordinates depend on the user's web browser settings and display dimensions. Additionally, the exact positions of the Gugubarra zones on the web sites have to be saved before it becomes possible to detect zone visits by the coordinates. Because of the high computational cost of the calculation for each user, we decided to not implement this approach.

The second set of stored meta data is the name of the highest level HTML tag in the DOM tree the mouse pointer is moved over. The major issue is the HTML code of the Gugubarra web site itself, i.e., the coding of the Gugubarra zones. In the HTML code of the web site a Gugubarra zone is represented by a div-container with the *class*

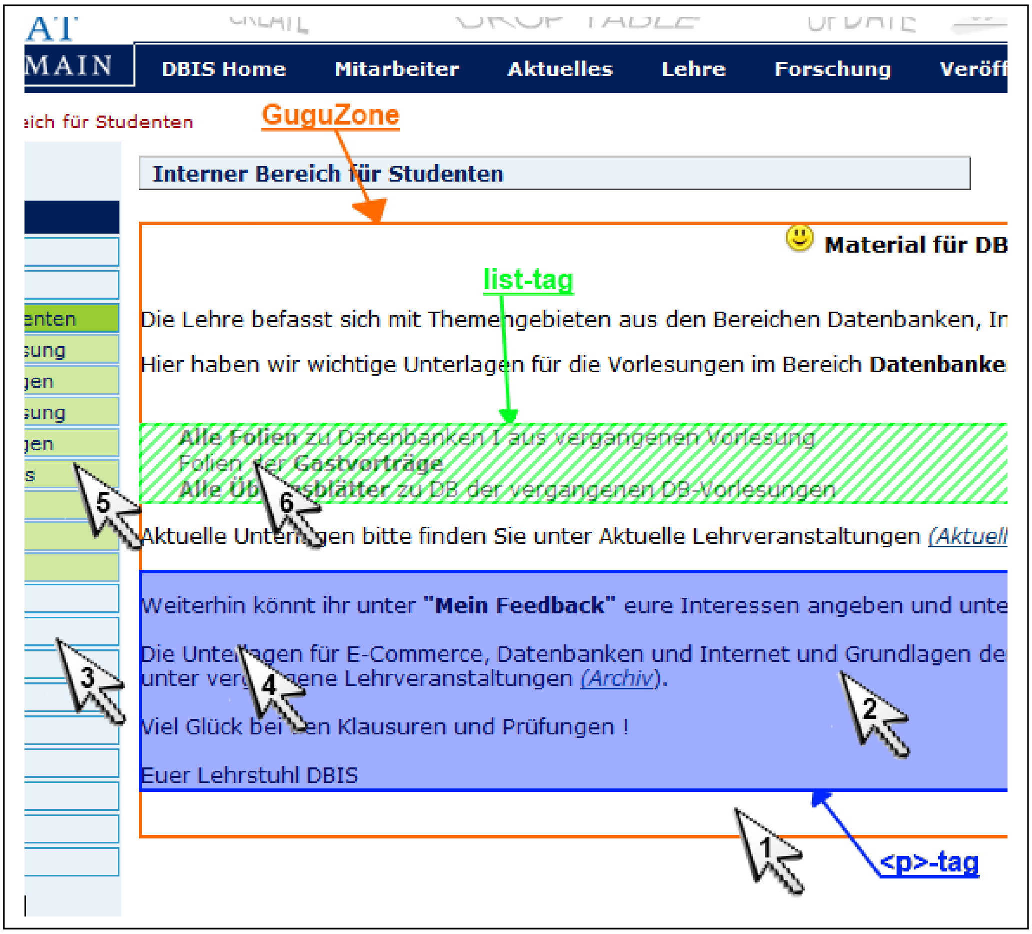


Figure 6.3: Mouse pointer in different Gugubarra zones

'guguZone' and the *id* of the zone. The following examples will illustrate this issue:

Let us assume a web site, as displayed in Figure 6.3, with *one* Gugubarra zone, some text elements, and some links. Figure 6.4 displays the HTML code of the Gugubarra zone. The mouse pointer at position 1 exhibits exactly onto the Gugubarra zone, so the HTML tag $< div >$, the class 'guguZone', and the id 102 of this zone will be saved in the session XML-file. This way, we can register one zone visit from the access to the leaving of this $< div >$ tag. However, if the user moves the pointer from position 1 to position 2 the meta data of this mouse trail would be: HTML tag $= < p >$, but *no* class and *no* id, because the Gugubarra zone id is defined in a higher level within the HTML tree of the page. As a consequence, we can not assess from the XML-file information that the mouse cursor is still in the same Gugubarra zone. The XML-file information displays the departure of the zone and the entrance of a new HTML tag with no zone information. For this reason, there is no way to register a complete zone visit.

```
<html xmlns=" ...>
  <body>
     ...
     <div class="guguZone" id="102">
       <ul>
         <li>...text ...</li>
         <li>...text ...</li>
         <li>...text ...</li>
       </ul>
       <p>...text ...</p>
     </div>
     ...
  </body>
</html>
```

Figure 6.4: HTML code of Figure 6.3

Another problem with the stored HTML information is shown by the mouse pointer movement from position 3, outside the Gugubarra zone, to position 4, which is placed inside the Gugubarra zone. The recording in the XML-file shows a leaving of the HTML tag outside the zone and the access of the HTML $< p >$-tag, but again no Gugubarra zone information.

Our first idea to overcome this issue by tracking the parental HTML-DOM tree element failed. In this approach, we store the parent element of the current HTML tag. As result, we are now able to find the $< div >$ element, which contains the zone information by reconstructing the mouse path through the HTML-DOM tree. For all previously described mouse moves, we can thus determinate the Gugubarra zone id because we always store the $< div >$ element with the Gugubarra zone id, which terminates our reconstructed path. In our example, the meta data of the mouse pointer movement from position 1 to position 2 are:

1: HTML $< div >$ tag, class="guguZone",
id="102", HTML-parent $< html >$ tag

2: HTML $< p >$ tag and HTML-parent $< div >$ tag

With this data, we are now able to reconstruct the Gugubarra zone information for position 2. However, this is not a satisfactory solution as can be seen from the movement, starting at position 5 and ending at position 6. Pointer position 5 is not within the boundaries of a Gugubarra zone, accordingly there will be no Gugubarra zone meta data written in the XML-file. The meta data at position 6 are: HTML $< li >$ tag and HTML-parent $< ul >$ tag. In this case, we can not trace back the zone visit because the HTML-parent tag is not the HTML tag of the previous mouse position.

Currently, it is neither possible to register all zone visits nor the duration of one registered zone visit. Nonetheless, registering zone visits by mouse-tracking still represents an extension in comparison to profiles without mouse-tracking information.

The same problem exists with memorizing mouse clicks. One mouse click on pointer position 1, shown in Figure 6.3, will be registered as a click on a Gugubarra zone. In contrast, mouse clicks on position 2, 3, and 4 will be registered merely as clicks without information about the Gugubarra zone. Analogously to the mouse pointer moves, it is not possible to register every zone id of all mouse clicks in Gugubarra zones during a session.

After analyzing all information tracked by the Gugubarra mouse-tracking module, the following information is used for the new mouse profile of a user: start and end timestamp of the session, number, type, and zone id (if possible) of the mouse clicks, zone id of the visited zones (if possible).

On the basis of these data, the new *Mouse Profile* (MP) is built, which is defined by the following formula:

Definition 6.4 *Mouse Profile (*MP*):*

$$MP_{u_m,t_n}(T_i) = \alpha * ZvP(T_i) + \beta * ClP(T_i)$$

According to the NOP, the *Mouse Profile* is calculated by the sum of two profiles, *Click Profile* (ClP) and *Zone visiting Profile* (ZvP). Each of them multiplied by a weight α, β. With these weights the web site owner can regulate the impact of the ClP and the ZvP on the Mouse Profile calculation.

The *Click Profile*, shown in Definition 6.5, calculates a profile based on the mouse clicks executed by a user during a session. Here, the profile distinguishes single clicks and double clicks. We create for both types weights 'a', 'b'. So the owner of the web site can attach different weights to both of them. For each zone Z_q with topic T_i, at first the count of single c_s and double clicks c_{db} multiplied by their weights are aggregated. After multiplying with the topic weight $v(T_i, Z_q)$, we sum up the result for all topics. Finally, the result is normalized by the sum of all counted clicks executed during one session.

Definition 6.5 *Click Profile (*ClP*):*

$$ClP_{u_m,t_n}(T_i) = \frac{\sum_q ((a * c_s(Z_q) + b * c_{db}(Z_q)) * v(T_i, Z_q))}{a * \sum_q (c_s(Z_q)) + b * \sum_q (c_{db}(Z_q))}$$

The following definition shows the calculation of the *Zone visiting Profile*, which takes into account the zone visits during a session:

Definition 6.6 *Zone visiting Profile (*ZvP*):*

$$ZvP_{u_m,t_n}(T_i) = \sum_q \frac{v(T_i, Z_q)}{\sum_{allvisitedzones} v(T_i, Z_q)}$$

The topic weight v for every single topic will be normalized by the sum of the topic weights of all visited zones during the complete session. For the result, we build the sum for all topics.

The Mouse Profile is now used to extend the Relevance Profile, which is defined in Chapter 3, Section 3.6 in Formula 3.2. The following definition displays the more accurate version of the RP:

Definition 6.7 *Relevance Profile with Mouse Profile:*

$$RP_{u_m,t_n}(T_i) = \frac{NOP_{u_m,t_n}(T_i) + f_i(S_i) * FP_{u_m,t_n}(T_i) + MP_{u_m,t_n}(T_i)}{a + b + f_i(S_i) + \alpha + \beta}$$

In contrast to the previous version of the RP (see Formula 3.2 in Chapter 3), we sum up the NOP, the FP, *and* the MP. In the end, we have to add the weights of the MP to the denominator.

6.6 Evaluation

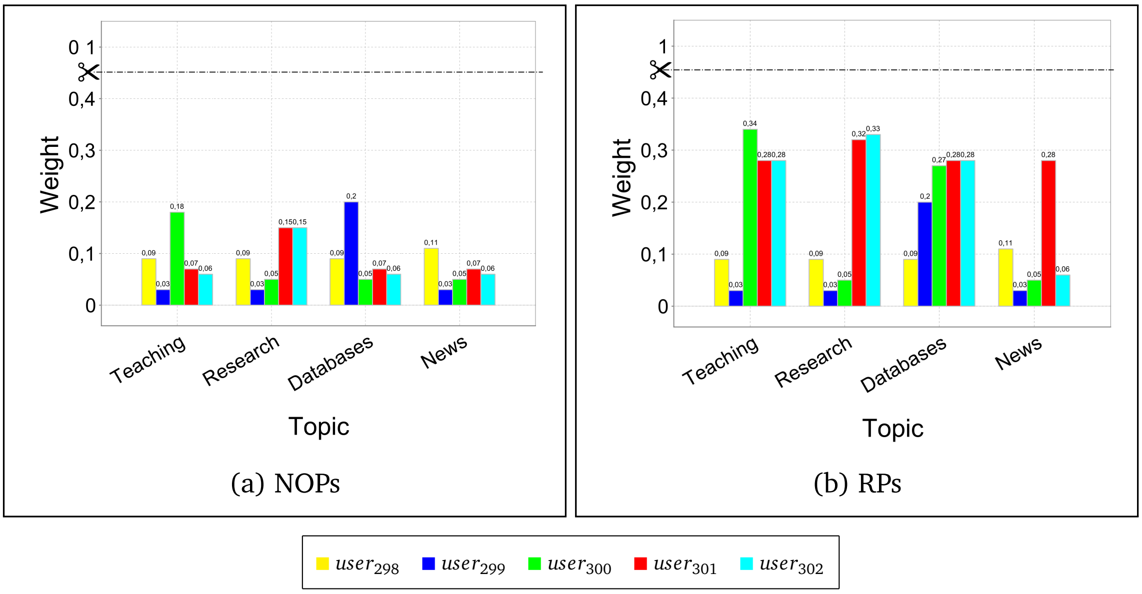
In this section, we evaluate the new mouse-tracking functionality of the Gugubarra Framework. Therefore, we conducted an experiment with two groups of users: the first group conducted the experiment with disabled mouse-tracking functionality, the second group with enabled mouse-tracking functionality. Next, we analyzed the user profiles of the two groups. Moreover, we compared the NOPs that contain no mouse-tracking data with the RPs that take mouse-tracking data into account.

6.6.1 Material and Methods

In order to test the mouse-tracking module of Gugubarra, we conducted an experiment with five users. All of the participants are coworkers of the DBIS research group, three female (60%) and two male (40%). We used the Gugubarra Framework, installed on the DBIS web site. Like in the case studies, described in Chapter 5, we used the following system settings:

- NOP with parameter $a = b = 0.5$:
  The DurP and the ActP had the same impact on the NOP calculation.
- We defined four topics for the web site (see Section 3.3, step 1):
  $T_1 = teaching$, $T_2 = research$, $T_3 = databases$, and $T_4 = news$.
- RP filter function constant with $f_l(S_k) = 1$ (see Section 3.4, step 2):
  With this setting the FP and NOP were equally important. However, during this experiment no explicit user feedback was collected because the FP was not the scope of this experiment and could have blurred the effect of the MP.
- MP with parameter $\alpha = \beta = 0.5$:
  The ZvP and the CIP had the same impact on the MP calculation.

For the experiment, we prepared a web page with four short preview texts (teasers) about current news with four different topics: teaching, research, databases, and general news. These teasers were put into zones, assigned with the appropriate topic and topic weight. The test procedure was as follows: the participants were instructed to read all news previews and select the most interesting one. If they selected an article they had to click on a "read more" link, which led to a page with the complete text of the news article. The length of the news article was chosen so that it did not fit completely to the computer screen, thus the participant had to scroll down to read the whole article. To ensure that the participant read the whole news article, we announced that they had to answer detailed questions about the content of the article afterwards. The experiment was supervised in a lab, so that all participants had the same conditions. During the experiment, the supervisor noted down the topic of the read article of every participant. The Gugubarra mouse-tracking module was enabled for three users ($user_{300}$, $user_{301}$, and $user_{302}$), without their knowledge and disabled for the rest ($user_{298}$, $user_{299}$) of the participants (control group).



(a) NOPs

(b) RPs

Figure 6.5: User profiles of the mouse-tracking experiment

6.6.2 Results

To evaluate the results of this experiment, we used the Gugubarra Framework. Figure 6.5a shows the NOPs, Figure 6.5b the RPs of the participants. On the y-axis the interest weight/value is displayed, on the x-axis the names of the different topics are shown. Each colored bar represents the NOP/RP value for one topic of one user. We collected *no* explicit feedback (FP) from the participants during the experiment. Therefore, the RP took only the mouse profile (MP) and the NOP into account. In contrast to the RP, NOP calculated the interests of the participants without making use of the data from the mouse-tracking. For the same reason, the NOP and the RP of the two users, which conducted the experiment with disabled mouse-tracking, should be the equal.

As seen from the notes of the experiment supervisor, the users read the articles with the following topics:

- Teaching: $user_{300}$
- Research: $user_{301}$, $user_{302}$
- Databases: $user_{299}$
- News: $user_{298}$

6.6.3 Discussion

The NOPs (see Figure 6.5a) of all users show a peak at the read topic. In general, the value of the weights are very low: the highest peak is at 0.2–on a scale with a range to 1.0, which is not a clear indication for an interest in a topic.

The RPs (see Figure 6.5b) also have a peak at the read topic. In comparison with the profile with disabled mouse-tracking ($user_{298}$, $user_{299}$), the profiles with mouse-tracking have higher weight values, but the difference between the read topic and the no-read topic is also very low. The cause could be the limitations of the mouse-tracking implementation, in respect to the zoning of Gugubarra: we were not always able to determinate in which zone the mouse of the users was because more accurate data from the zones for the calculation were missing.

In conclusion, the mouse-tracking data provide more valuable information about the visited web pages and about the interests of a user. But with the limitation in monitoring Gugubarra zones, the mouse-tracking is not as useful as it could be. Our implementation still needs adjustments in the observation of the Gugubarra zones.

6.7 Conclusion and Future Work

The implicit user data, extracted by the Gugubarra mouse-tracking module, combined with the other user interest profiles, allows us to generate more accurate interest profiles of the users. We have proven our concepts in an experiment and shown that the mouse-tracking data improve the accuracy of the user interest profiles. This gives a solid basis for prediction of user interests or trend detection in the Web. Since implicit profiles are less prone to manipulation, they provide a chance in improved personalized services, including, but not limited to, advertisements, while allowing better detection of trend changes and their adoption. We will test our new concept in a forthcoming user study with a larger quantity of participants. It would also be interesting to observe with the help of mouse-tracking web site visitors that are using a device with a touch screen, like a smartphone or a tablet computer. These types of devices become more and more popular and should be considered as source of user behavioral data.

In future, we plan to improve our mouse-tracking module prototype with the help of new technologies. As the new HTML5[6] standard promises to introduce a multitude of exciting new features, we expect to further improve the accuracy of our tracking based mouse-profiles. Especially the upcoming user attributes feature seems to be suitable to achieve such improvements.

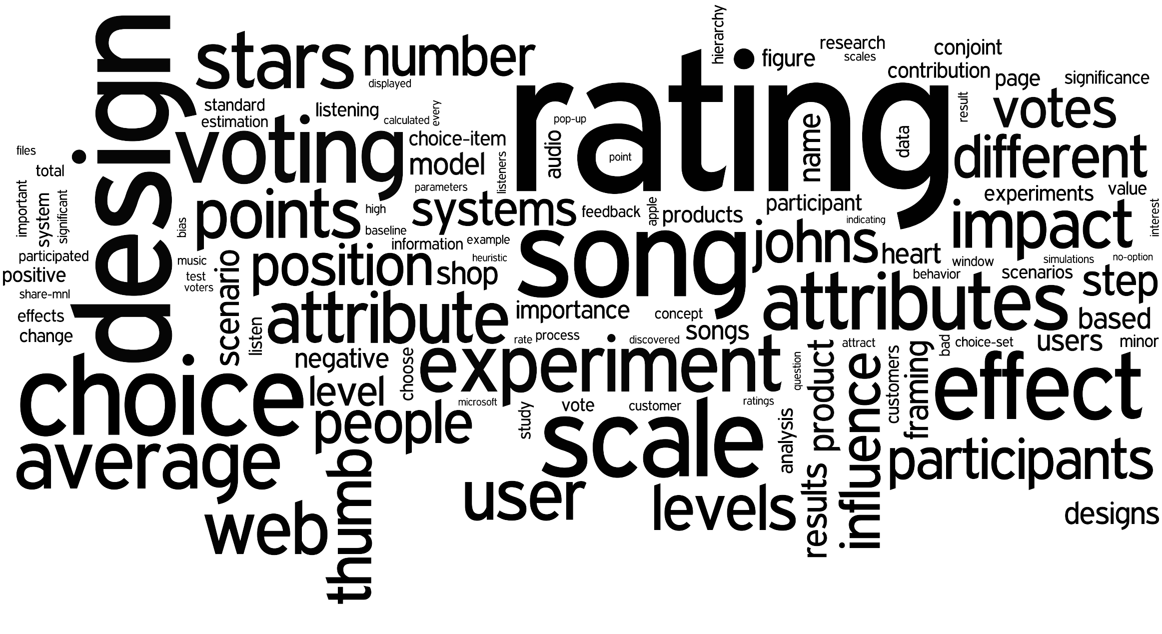
[6] http://dev.w3.org/html5/spec/

Table 6.1: Contribution summary

Subject	***Detail***
Mouse-event e_i	Any action performed with the mouse (Definition 6.1)
Mouse trail T_i	A sequence of mouse-events (Definition 6.2)
Guguboomla! Mouse-Tracking	We designed and implemented a Joomla! extension for the Gugubarra Framework to track the mouse actions of a web site visitor (Section 6.5.1).
Mouse Profile (*MP*)	We introduced a new user interest profile, which calculates the interest with the help of the mouse-tracking data. It consists of the Click Profile (CIP) and the Zone visiting Profile (ZvP) (Section 6.5.2).
MP integration into the RP	We extended the Relevance Profile (RP) with the new Mouse Profile (MP) (Definition 6.7).
Proof of Concepts	We proof the accuracy of the mouse-tracking extension of Gugubarra with a user experiment (Section 6.6).

Part II

Influencing Factors on Web User Feedback



7 How Do Web Based Rating Systems Influence User's Choice

To understand the complete process of web user feedback, different factors that influence the user's choices on giving feedback are investigated. With the knowledge about these factors, web business owners will be able to collect suitable feedback for the analysis. To measure these factors, we conduct three experiments on a real web store. This chapter is based on the publication of Schefels et al. [SWHZ13].

7.1 Introduction

With the rise of the Web 2.0 [O'R05], users' collaboration and feedback became more and more important. For this reason many web shop owners provide their customers the possibility to rate or comment on the products of the shop. For example, the web store Amazon[1] has a rating system and a comment field for every product. The customer, while searching for a product, often has the possibility to rank the products according to the users' rating scores. Even the recommendation system of the shop takes advantage of these user ratings and suggests high-rated products to the customers.

[1] `https://www.amazon.com/`

Not only the shop owner benefits from these ratings but also the consumers do: high-rated products often meet the expectations of the consumers [WO81] while low-rated products often fail to satisfy them. The advantage for the customers is also an advantage for the shop owner: he saves money and time because his customers complain less.

However, there is also a misuse of rating systems. Companies try to rank their products high by cheating (comment spam) [HLS11] and users may use the systems only for fun.

Nevertheless, user rating systems are a benefit for web shop owners and their customers [GI07].

Another set of Web 2.0 phenomena, which became more and more important, are social networks like Facebook[2] or Google+[3]. In these networks, users can rate or recommend products or web pages to their friends as well. For web shop owners, it is an easy and powerful way to reach millions of customers. For example, Facebook and Google+ designed special buttons that can be easily integrated into every web page/shop to make recommendations to the whole community. A web shop customer or a member of a social network has only to click on such a button and all his friends see his recommendations (in some case even the whole network).

Until now, a lot of research has been done to investigate the effect of the ratings of products with the result that user ratings are important for the popularity and the sales volume of products. Our focus is on the different designs of the rating scales [CM06]. In the Internet, a wide variety of different rating systems can be found, e.g., Amazon's common five star product rating system or the very popular "thumb-up" recommender symbol of Facebook. Accordingly, our research question is as follows:
Do these different styles of rating scales have an influence on the willingness of the user to give feedback or are there other influencing factors?

We split this research question into three parts and focus on web rating systems in on-line music stores:

- *How does the representation design of the community feedback information influence the user's choice to listen to an audio file?*
- *How does the representation design of the community feedback information influence the user's choice to rate an audio file?*
- *How does the representation design of the community feedback information influence the user's choice to indicate interest in an audio file?*

[2] https://www.facebook.com/
[3] https://plus.google.com/

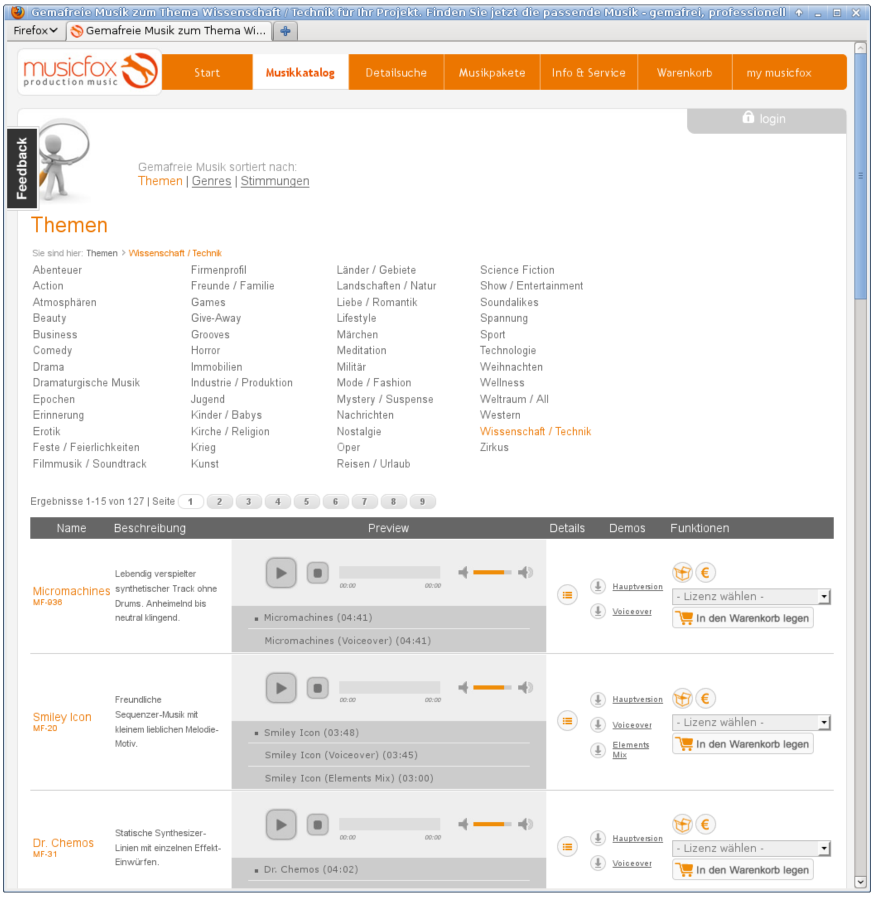


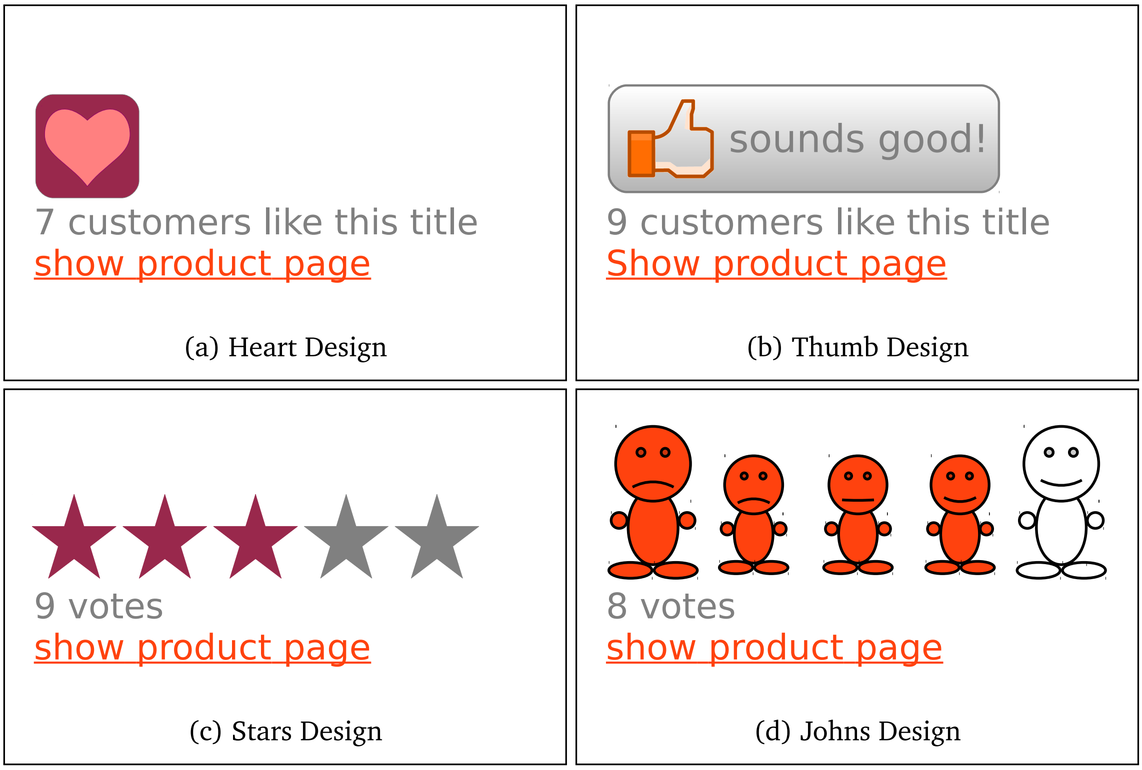
Figure 7.1: The musicfox web shop

The experiments took part on the musicfox[4] music store (see Figure 7.1), a web shop for "gemafreie[5]" music.

The rest of the chapter is structured as follows: Section 7.2 discusses the theoretical insights and introduces a cognitive process of rating. Section 7.3 presents the method and the research design of our study. In Section 7.4, we present four steps are presented, which allow us to evaluate the conjoint model. We use these four steps in Section 7.5, Section 7.6, and Section 7.7 to resolve our three research questions. Section 7.8 discusses the results of the three conducted experiments and Section 7.9 presents the conclusion and future work.

[4] `https://www.musicfox.com/`

[5] *GEMA*: A German acronym for the society for musical performing and mechanical reproduction rights (German: **Ge**sellschaft für **m**usikalische **A**ufführungs- und mechanische Vervielfältigungsrechte). "*Gemafreie*" (GEMA-free) music: the music producer is *not* a member of the GEMA and can sell his music under his own licenses. Therefore, he has not to pay a fee to the GEMA.



(a) Heart Design (b) Thumb Design (c) Stars Design (d) Johns Design

Figure 7.2: The different rating scale designs

7.2 Theoretical Insights

In the rest of the chapter, we use the term "rating scale design" to describe the graphical interface for a rating system. Figure 7.2 displays all rating scale designs we will use in our experiment; for further details see Section 7.3.1 (The Research Design). However, in other publications, rating scales are also referred to as "scoring system", "voting system", and, in general, as "feedback mechanism".

Nowadays, there is a wide variety of rating scales with different designs. A very common scale is the Likert scale, which has a certain number of levels. For example, a participant could express her/his agreement by a ten-level scale where zero means "total disagreement" and ten means "total agreement". Today, in times of extensive usage of the Internet, rating systems are often realized by so-called HTML radio buttons in web forms. Besides radio buttons, slider scales are used with no certain levels. The user can move the slider from a minimum level to a maximum level. The levels between the two limits have almost an infinity of fine granularity and are, therefore, difficult to match to their semantically equivalents. If fewer details are needed, the use of binary rating systems with two states is reasonable. For example, the on-line video watching and sharing web site Youtube[6] uses a rating scale with only two levels, "like" or "dislike". The simplest rating system has a unary scale with only one level. Today such a kind of scale is very common through the popularity of the on-line platform Facebook that makes usage of this scale form. The Facebook unary rating scale has a "like" button. With the help of this button a Facebook member can express her/his favor to a comment or to a web site and show it to the Facebook community. Unary rating systems allow no negative rating and can be useful for shop owners who want to avoid negative feedback.

Before the utilization of the Internet became common, the main area of application of rating systems was opinion polls. In contrast to surveys, where the granularity and meaning of the single levels of the scale has to be explained to the participants, rating systems of products do not usually provide such an explanation. Product rating systems often present a short statistical overview to the user with the information of the rating behavior of the former users (e.g., how many people gave their vote or the average voting points of all former voters). This is not reasonable in surveys because this could influence the answers of the participants.

Many researches on the usage of rating scales in surveys or economics discovered a j-shaped distribution of the votes (e.g., as described in [HZP09] by Hu et al.). However, we are not interested in the matter how people vote. Instead, we want to clarify the influence of the different factors that affect the voter during the rating process.

In economics, consumer reviews are often seen as a new kind of word-of-mouth (WOM) [HPZ06, Del03, CM06, XB08] propaganda, sometimes called e-WOM [SL07]. These reviews express the experience of the customer with a product and frequently include suggestions to other customers interested in the same product. Consumer review systems consist usually of a text-form for user comments and a rating scale. In this chapter, we are concentrating on a pure rating scale without the possibility to post text comments. A certain form of the WOM are reputation systems. For example, the rating system of the on-line auction portal eBay[7] is used to rate the trader after a transaction. The more positive ratings the trader earns for her/his former transaction, the greater is the community's trust in her/his reputation. Dellarocas presents in [Del03] an overview focusing on rating systems within recommender systems. However, we do not use the collected votes of the customer for recommendations, but we are only providing the rating system to the customer in order to value a product, i.e., a song. How the collected feedback is used, is not part of this study.

Accordingly, from an economic perspective, rating can also be seen as an economical process, where the user wants to benefit from her/his action. In [HLCK05], Harper et al.

[6] `https://www.youtu.be/`
[7] `http://www.ebay.com/`

built a parameterized economic model of rating behavior to explain the motivations of the users. The authors suggest a personalized rating interface to enhance the motivation of users to rate.

Xia and Bechwati detected in [XB08] that the impact of reviews depends on the environment. This is explained by the affect intensity, which means how emotionally people respond to different events. In this research work, we also observe the environmental attributes that might influence the voter in her/his actions. In detail, we include in our observations different attributes from the design of the rating scale as well as from the design of the web page of the experiment. We also refer the aspects of emotions in our discussion of the results.

Sometimes the effect of product ratings is classified by the product category: Sen and Lerman discovered in [SL07] that negative ratings for hedonic products are not as helpful as for utilitarian products, i.e., they did not observe a negativity bias for hedonic products, but a positivity bias. Similar to this discovery Xia and Bechwati explored in [XB08] that the impact of negative ratings of experience/hedonic products on people with high affect intensity is higher than the impact of positive ratings. In this work, customers of a web shop have to rate songs, which can be put into the product category of hedonic products. So, we expect similar effects on negative ratings.

In [RBLK10], Riedl et al. assume that the cognitive process of rating ideas in open innovation platforms is very similar to the one responding to a survey. This response process is analyzed in detail by Tourangeau et al. in [TRR00] and a four-component model is suggested. As we are dealing with the rating for audio songs, we have to adopt the model to our needs, as follows:

Comprehension: in a survey the attendant has to read, understand, and assign a meaning to the surface form of the question, as Tourangeau et al. observed in [TRR00]. This is very important because misunderstood questions could confuse the attendant. So, she/he might answer in a way different from what was intended by the interviewer as Collins shows in [Col03].

Therefore, since we have neither a survey with questions nor attendants, we have to adapt this component to our case: first the customer, considered a sort of attendant, has to click on the link to open the experiment web page and understand all information presented in it. The information consists of all available attributes (i.e., the visible attribute levels, see Table 7.1) and/or listen to one or more songs. If a customer does not understand the meaning of this experiment web page, he will simply close it without rating any song. That is the cause why we think that the design of the rating scale is very important and leads us to our hypothesis.

Retrieval: after understanding the question and its meaning, the attendant will recall relevant information related to this question from his long-term memory [TRR00]. For example, Sloboda discovered in [Slo92] that songs can trigger positive, as well as negative emotions (shown by Baumgartner in [Bau92]). Thus they can influence the rating behavior of the customer because some people connect memories with songs. These memories are kept in the long-term memory and have to be recalled. All other attributes are equally important and should be considered in every research.

Judgment: in this component, the attendant forms her/his answer about the question [Col03]. Here, the attendant will decide whether enough and sufficient memories, recalled during the Retrieval, are available to answer the question. If the question is difficult, the attendant has to draw conclusions, to infer missing details, or to combine fragmentary memories to answer the question. Difficult are also questions about date or frequencies because often it is difficult for people to remember exact date (see Friedman and Wilkins [FW85]).

Judging songs is similar to judging attitude questions. The judgment can not be based on facts; it is a very personal decision. Collins presents in [Col03] "Judgmental heuristics" that are often employed when attendants respond to a question. Here we will list some well-known heuristics that can be put into context to judging and rating songs:

- *Affect heuristic*: positive as well as negative emotions influence the choice, as Finucane et al. show in [FASJ00]. We expect that emotions will have an influence on the user choice. For example, the appearance of the manikin in the rating scale design "Johns" (see Figure 7.2d) with its large head, large eyes, and round body could be seen as a scheme of childlike characteristics, i.e., cuteness. Therefore, we expect that the voters will be attracted by this design in particular.

- *Anchoring*: people tend to use an initial value [TK74] as a basis point for their estimation. As a result, with different initial values people will estimate different results. We expect that this effect influences the choice of the users if they see a five-point rating scale (Johns or Stars design) with the average voting points of the last voters.

- *Availability heuristic*: is based on the fact that the choice depends on memories that are available from past experiences. Tversky and Kahneman describe in [TK74] several studies where people making assessments often are misled by their biases.

- *Bandwagon effect*: people prefer choice-items that are chosen by the majority. In [MJ09], Margetts and John discovered that the willingness of people to sign an on-line petition increases by the information that there are other signatories. We expect the influence of this effect if we show the users the total number of votes. Bond et al. present in [BFJ$^+$12] an example of the Bandwagon effect on Facebook, where American voters are influenced in their choice by the rating results of the community.

- *Framing effect*: the design of the experiment web site may affect the choice of the participants, described in [HSDA07] by Hartmann et al. Even emotions, as De Martino et al. show in [DMKSD06], are associated with the framing effect–listening to songs may also trigger emotions.

- *Negativity bias*: in [Kan84], Kanouse shows that people weigh negative information more seriously than positive information. Therefore, we expect, that negative ranking of songs could influence the choice of the participants.

- *Positivity bias*: Sen and Lerman observed in [SL07] a positivity bias in correlation with rating of hedonic products. It is the opposite of the negativity bias, which means that people weigh positive information more than negative information.

- *Serial position effect, i.e., Primacy effect*: in [MHM06], Murphy et al. report that web users prefer a higher ranked link in a list of links, which is called primacy effect. Tourangeau et al. discovered in [TCC04] a similar effect: in an ordered list of categories most people expect that the top item is the most desirable one. Tourangeau et al. call the effect "up means good".

- *Serial position effect, i.e., recency effect*: in [MHM06], Murphy et al. point out as well that the last element in a list is more important than the elements in the middle of a list. Therefore, we expect, that the rank of the positions of the three choice-items will be: first position (primacy effect), third position (recency effect), and second position.

Most of the researches, and thus the research results, are related to surveys or questionnaires. The aim of a survey is to get the opinion of the participant without any influence caused by the researcher. Therefore, the researches try through surveys to discover how to minimize this effect by recognizing various heuristics about decision making. Our research aims at the opposite direction: we want to discover how to influence people to motivate them to vote more for a product. This has advantages for the customer as well as for the web shop owner, as pointed out in the introduction.

Response: finally the attendant has to map the answer onto the appropriate scale or response option [TRR00]. This means, in our case, that the customer, who wants to vote for a song, has to map his judgment to the provided rating scale. He can also express his attitude to the song by visiting the product page of the song, which we interpret as an intention to buy the song.

7.3 Research Method and Design

7.3.1 The Research Design

For our study, we used the musicfox web shop to collect data. The participants are *both* the walk-in customers and the registered customers of the musicfox shop.

We prepared a pop-up web page with three different songs (see Figure 7.4) and placed it on the home page, the so-called landing page, of the shop. To draw the attention to the link and to arouse visitors' curiosity we named it "*top songs*" and placed an eye-catching icon next to it. The web page of the pop-up window contained a player for each song so that the songs can be listened to on-line. Next to every song a rating system with its scale was placed. The three rating scales always had the same design per IP-address so that a single participant only saw *one* kind of the rating scale design. The design altered randomly for the visitors, i.e., for the IP-addresses.

We designed *four* different rating scales: a heart (unary), known from Last.fm[8], a thumb-up Facebook-like rating scale (unary), an Amazon-like rating scale with five stars (Likert scale), and a new style inspired by SAM [BL94], a non-verbal pictorial assessment technique to directly measure emotion (Likert scale). All styles are shown in Figure 7.2. Sparling and Sen discovered in their paper [SS11], that users prefer the five stars design as well as the thumb design for product reviews. For this reason, we created two five-star-like (see Figure 7.2c and 7.2d) and two thumb-like rating scales (see Figure 7.2a and 7.2b).

Every visitor had the possibility to listen to the songs, to rate the songs, and to click on a link, which opens the product page of the song in the musicfox web shop, where she/he could gather information or buy it. During the experiment, when the participant closed the pop-up window without performing any action, the no-option (see Section 7.4.2) was selected.

We performed three experiments through which we observed different types of behavior of the participants: in the first experiment (*E1*) we explored the impact of service attributes on the choice of listening to audio files, in the second experiment (*E2*) we

[8] `http://www.last.fm/`

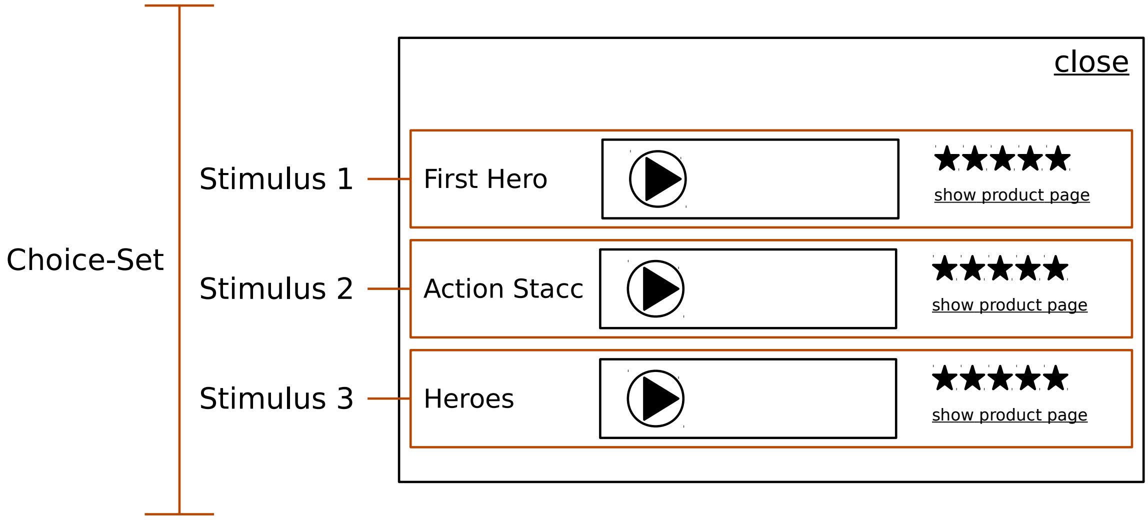


Figure 7.3: Choice-set with the five attributes per stimulus

explored the impact of service attributes on the choice of rating, and in the third experiment (*E3*) we explored the impact of service attributes on the choice of indicating interest toward a product (song).

We sent no invitations to people to ask for participation in the experiment because we did not want to encourage the participants to use the rating system–they should act as naturally as possible and only use the rating system if they want to. Without knowing about the experiment, the participants were not affected by the Hawthorne-effect [FK78], so the result should be unbiased.

7.3.2 The Choice of Attributes

To identify the relevant attributes, we have been investigating the most popular music web site, which makes use of rating systems. Our choice-set, the pop-up window, consists (see Figure 7.3) of three stimuli with five attributes per stimulus. Table 7.1 summarizes all attributes, as explained below:

A choice-set contains three different stimuli, each with one song (see Figure 7.3). Every song has a different name that is displayed on the left side of the music player. We chose three unknown songs with very generic names; therefore, the attribute "Song Name" has three different levels: First Hero (mf-78), Action Stacc (mf-170), and Heroes (mf-297). We chose very general names on purpose, on the one hand to make the participants curious about the song and lure them to listen to it. On the other hand, we wanted to avoid that participants vote for a song with a well-known name without

Table 7.1: Attributes and attribute levels

Attribute Name	Number of Levels	Levels
Song Name	3	mf-78, mf-170, mf-297
Song Position	3	1, 2, 3
Rating Scale Design	4	Johns, Stars, Thumb, Heart
Number of Votes	100	$[0, 100]$
Average Voting Points	5	0, 1, 2, 3, 4, 5

listening to it.

In the choice-set, the three songs are placed in a row; therefore, the attribute "Song Position" has three levels: a song could be placed on the top, in the middle, and at the bottom of the list of songs. The songs were ordered randomly for each participant.

A crucial role is played by the attribute "Rating Scale Design". We created four different designs for the rating scale: Johns, Stars, Thumb, and Heart. While Johns and Stars are rating scales with five levels, Thumb and Heart only have one level. Stars are well known from various web sites like Amazon or Musicload (`https://www.musicload.de/`). Johns are a new creation inspired by SAM, a pictogram used in psychology to display emotions [BL94]. Thumb is also well known through the popularity of the social network Facebook. Heart is used by Last.fm and often related to the action to favor something (i.e., "I love it.").

We recognized during our investigations on rating systems that many web pages display, next to the rating scale, the current number of the people that voted for a song. For this reason, we placed a counter with the current number of people that voted for this song next to the rating scale. This attribute, called "Number of Votes", is a linear attribute, which means that it has an infinitive number of levels.

For the rating scale designs with more levels, the Johns and the Stars, we also displayed the average voting points for the participants to see. This average is calculated by dividing the total number of voting points by the number of votes. The result is between zero and five, since the result of the division has to be rounded to the next whole number. The result is visualized by coloring the pictograms of the rating scales differently: for example, if the average voting points are three stars for a song, the three first stars of the rating scale design are colored in red, see Figure 7.2c. The unary designs Thumbs and Hearts have no different levels, for this reason this kind of visualization is not possible here.

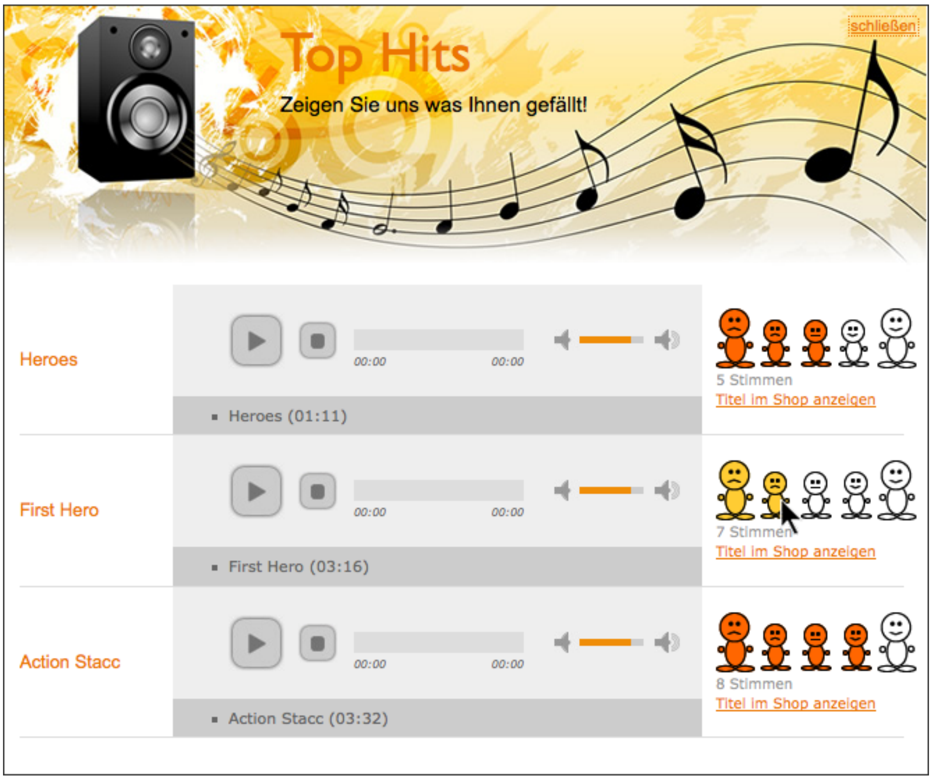


Figure 7.4: The pop-up window of the experiments

7.3.3 Research Design and the Research Instrument

With this study, we wanted to see how much the attributes, introduced in Section 7.3.2, influence the choice of the user. All attributes were combined in one stimulus, see Figure 7.3, and three stimuli per participant were presented in one task. One participant saw only one rating scale design. The position of the song, i.e., the position of the song name, was selected randomly, while the number of average voting points were calculated from the previous voting data. The design of the experiments was calculated with SAWTOOTH[9].

We kept the layout of the pop-up window minimalistic. We discovered the presented attributes in an Internet investigation. We selected only attributes that are in common at most of the music shops. The layout of the pop-up window, see Figure 7.4, was the same used in the musicfox web store. This has been maintained to make the participant feel comfortable. Besides, the participants do not recognize the real purpose of the pop-up window, i.e., the case study. The study was implemented with the self-developed PHP framework of the musicfox web store.

[9]Sawtooth Software, Inc., http://www.sawtoothsoftware.com/

7.4 Analysis Overview

Since the 1970s, the use of the conjoint analysis has gained in popularity with both academics and practitioners [GS90]. We use the choice-based conjoint (CBC) analysis for the purpose of this study. In [Hoe11], four steps are presented, which allow us to evaluate the conjoint model:

- Step 1: Choice and definition of the estimation model for all experiments.
- Step 2: Estimation of the effects using the estimation model and test of the significance of the effects and of the Standard Conjoint Model.
- Step 3: Calculation of the importance hierarchy and the contribution size of the attributes.
- Step 4: Simulations of different choice tasks.

Step 1 is the same for each of the three experiments (E1, E2, and E3) and presented in Section 7.4.1.

Steps 2 to 4 are separately calculated for each experiment. The common parts of steps 2 to 4 are presented in the subsequent sections. The results of these steps are presented for each experiment in a separate section; E1 in Section 7.5, E2 in Section 7.6, and E3 in Section 7.7.

7.4.1 Step 1: Choice of Estimation Model

First, we chose a parameter estimation model to use for the calculations. This step is the same for all experiments E1, E2, and E3.

There are different ways to analyze the results of a choice-based conjoint study. This section describes the analysis on the aggregate level, using the multinomial logit model (MNL). MNL is based on the work of Mcfadden [Mcf74]. For the coding of the data, we used effect coding, which is compatible with SAWTOOTH Software. With effect coding, the sum of all effect parameters of the levels of one attribute will be equal to zero; for further details, see [Coh03]. The multinomial logit model is defined by using notations adapted from Backhaus et al. [BEPW05].

Adoptions are as follows: We changed some variables for better readability. For example, instead of using β we use $\beta_{a,l}$ to make it clear that an effect exists for each level l of an attribute a and instead of using k for a concept (in this study a song with its rating system) we use c.

We first calculated the utility of each concept (pop-up window). The utility represents the attractiveness of each concept. The utility of a concept is calculated by summing the parts that each attribute level (shown in this concept) contributes. In other words, the utility of a concept c is given as the sum of part-worth utilities of each level l for each attribute a that is in concept c:

$$utility_c = \sum_a \sum_l \beta_{a,l} * x_{c,a,l} \tag{7.1}$$

where

- $x_{c,a,l}$ is the effect coding variable value of level l for attribute a for the concept c,
- $\beta_{a,l}$ is the effect parameter (part-worth utility) of level l for attribute a, which has to be estimated.

This results in a utility for the no-option [HKW01] that is equal to the effect parameter of the no-option: $u_{no-opt} = \beta_{no-opt}$. We then calculate the probability that a concept c is chosen:

$$prob_c = \frac{e^{utility_c}}{\sum_{c' \in CS} e^{utility_{c'}}} \tag{7.2}$$

where

- $e^{utility_c}$ is the exponent of the utility of this concept,
- $\sum_{c' \in CS} e^{utility_{c'}}$ is the sum of exponents of the utility of all concepts c' shown within the choice-set CS (in this study, a combination of three items and the no-option).

We used the *log-likelihood function* [BEPW05] to estimate the effect parameters $\beta_{a,l}$ by maximizing the value of the following function:

$$LL = \sum_{CT} \sum_c \ln(prob_c) * choice_{CT,c} \tag{7.3}$$

where

- $choice_{CT,c} = \begin{cases} 1, \text{if } c \in CS \text{ was chosen by a participant in a choice task } CT \\ 0, \text{ else} \end{cases}$
- CT is a choice task, i.e., the participant has to choose a concept (item) out of a choice-set CS.

The aim is to define a function that estimates the parameters $\beta_{a,l}$ in such a way that the observed user choices can be explained plausibly. This means that the probability a concept is chosen $prob_c$ should be as high as possible for all concepts that were, in fact, chosen by the participants. The function we use here, is the *log-likelihood function*.

The log-likelihood function calculates the sum of the log value of the probability for each c that was chosen. The value is added for each time the concept c was chosen. The function is calculated for different values of the parameters $\beta_{a,l}$. Finally, the parameter values that result in the maximum log-likelihood value (LL) are chosen.

The results of applying the defined logit analysis on the data of the study and the test of significance are reported in step 2.

7.4.2 Step 2: Estimation of the Effects and Test of the Significance

Step 2 has been separately performed for the data of each experiment (results see Sections 7.5.2, 7.6.2, and 7.7.2). The estimations were done for the standard conjoint model that includes all main effects according to Iyengar et al. [IJK08], with a degree of freedom $DF = 10$ (details on DF calculation see, e.g., Constantiou et al. [CHZ11]). The main results of step 2 for each experiment are shown in Tables 7.2, 7.8, and 7.14.

The effect parameters (Part-Worth Utilities) are listed in column three of each table for the attributes and their relative levels as presented in Table 7.1). We calculated the effects for all attributes, namely one effect for each level, and in addition one effect for the no-option.

The attribute no-option indicated that none of the above attributes were selected. Instead, the participant closed the pop-up window and did not select any option of the experiment.

Column three of each table presents the resulting effect parameter values, which are a measure of relative worth. The higher the value is, the more attractive the level is. The effect parameter values for the attributes “Song Name”, “Song Position”, and “Rating Scale Design”, is zero-centered. Therefore, the sum of all effect parameters of the levels of one attribute is zero. The attributes “Number of Votes” and “Average Voting Points” are linear; a minus indicates that the higher the value is, the less attractive it is.

The standard errors are shown in column four of each table. The standard error [WBW90] of the estimation is a measure of error in the prediction of the user’s choice. The t-ratio is shown in column five, which is the ratio of effect and standard error. This value can be compared with a t-table, taking into account the degree of freedom $DF = 10$, to determine the significance of each level. This significance level is marked in the third column with an asterisk.

In step 4 we further explore these findings by simulating different choice scenarios.

7.4.3 Step 3: Calculation of the Importance Hierarchy

Step 3 has to be separately performed for the data of each experiment to determine the importance hierarchy (results see Sections 7.5.3, 7.6.3, and 7.7.3).

The *attribute importance hierarchy* (*IH*) can be calculated on the basis of the estimated effect parameters of step 2. The results are shown in Tables 7.3, 7.9, and 7.15 and the figures accordingly. The importance hierarchy is indicated in the first column and the attribute importance in the third column. The attribute importance is an indicator of the influence size an attribute may have.

Column four highlights the significance level (contribution), that was calculated with a likelihood ratio test to determine whether the contribution of each attribute to the model was significant (details on the method see, e.g., Constantiou et al. [CHZ11]). Thus, *not* significant attributes have a minor effect on the user's choice; see also results of step 4.

7.5 Experiment 1 (E1): Exploring the Impact of Service Attributes on the Choice of Listening to Audio Files

7.5.1 The Sample of E1

A total sample of 117 people participated in the experiment 1 (*E1*), whereby 104 (89%) participants listened to at least one track and 13 (11%) closed the experiment window without any further action (no-option).

Four people (3%) participated four times in the experiment, two people (2%) participated three times, three people (3%) participated two times, and 99 people participated just one time, whereby participating means opening the experiment window.

Most of participants, 108 people (92%), took part in the experiment from Germany, one from Austria, one from Iran, and one from Switzerland. The location of six participants could not be determined.

The majority of the participants, 89 people (76%), used Microsoft Windows as the operating system to run the experiment. Apple Mac OS was used in 24 cases (20%), Linux three times (3%) and Apple iPad two times (2%). Note that one person used Microsoft Windows one time and Apple Mac OS another time to participate the experiment.

Most of the participants, 62 people (53%), used Mozilla Firefox as a browser, Microsoft Internet Explorer was used by 21 people (18%), Google Chrome by 18 people (15%), Apple Safari by 13 people (11%), Apple iPad by two people (2%), and Opera by two people (2%).

These data were all extracted from the server logs, the participants were not asked about any demographical data. However, in [TCC07] no impact of gender, age, or education groups by the use of rating systems is measured.

7.5.2 E1 - Step 2: Analysis Results of the Estimation of the Effects and Test of the Significance

Step 2 has been performed for the total data of this study, i.e., of 117 respondents, in total 429 choice-sets. The results of step 2 for experiment 1 are shown in Table 7.2.

The LL value is -527.07 for the standard conjoint model, using the estimated effects of Table 7.2. The likelihood ratio test was used to calculate the significance of the standard conjoint model. The chi-square value was 163.00 (degree of freedom $DF = 10$). The standard conjoint model is significant at the 0.0001 level.

Table 7.2: E1: Effect parameters

Attribute Name	Levels	Effect	Std. Error	t-Ratio
Song Name	mf-297	+0.01205	0.06936	+0.17374
	mf-78	−0.10358*	0.12905	−0.80263
	mf-170	+0.09153*	0.13017	+0.70315
Song Position	1	+0.23102***	0.06678	+3.45943
	2	−0.11592*	0.07210	−1.60774
	3	−0.11510*	0.07214	−1.59551
Rating Scale Design	Stars	+2.04826**	0.87604	+2.33808
	Heart	−1.05546*	0.61454	−1.71748
	Thumb	−1.51248**	0.58667	−2.57807
	Johns	+0.51968*	0.62157	+0.83606
Number of Votes		+0.01460	0.06575	+0.22202
Average Voting Points		−0.32417**	0.16709	−1.94008
No Option		−1.87561***	0.37255	−5.03448

* Significant at the 0.5 or 0.2 level.
** Significant at the 0.1 or 0.05 level.
*** Significant at the 0.01 or 0.001 level.

7.5.3 E1 - Step 3: Analysis Results of the Calculation of the Importance Hierarchy

According to Table 7.3 and Figure 7.5, the "Rating Scale Design" is the most important attribute (60.51%), followed by "Average Voting Points" (27.55%), and the "Song Position" (5.90%). The attributes representing the song name and the number of votes have only a minor contribution and influence on the choice of listening to audio files.

7.5.4 E1 - Step 4: Simulations of different Choice Tasks

After the analysis of the results, we are now able to perform several simulations to discover the influence of the single attributes in different scenarios. A scenario is a certain combination of attributes with a high contribution level to the model and it represents

Table 7.3: E1: Importance hierarchy and significance level for attribute contribution

IH	Attribute	Attribute importance	Contribution level
1	Rating Scale Design	60.51%	0.001
2	Average Voting Points	27.55%	0.001
3	Song Position	5.90%	0.01
4	Song Name	3.32%	minor
5	Number of Votes	2.73%	minor

a choice-set shown to a participant. In contrast to the choice-sets, we presented in the experiments, we are not longer bound to statistical or logical constraints: we are now able to combine, e.g., all different rating scale designs in one choice-set or put all songs on the same position in a simulation. Therefore, we can calculate the ratio of the choice from various combinations without testing them in reality. With these simulations the different impacts of the single attributes become much clearer. In the tables, gray printed entries indicate a change in the attribute level.

The first scenario of the first experiment combines different levels of the three attributes "Position of the Song", "Rating Scale Design", and "Average Voting Points". In this scenario, we took the attribute levels with the smallest impact, also called bad framing. Therefore, we put the song on position two, use the "Thumb" rating scale design, and display an average voting point of five to the participant. To discover the impact of the different designs, we change only the designs in this scenario from the design with the lowest impact ("Thumb") to the one with the highest impact ("Stars"). As a result, 77.5% of the participants would choose the choice-item with the stars rating scale design to listen to, only 2% choose the item with the thumb design. The detailed results are presented in Table 7.4. The changed attributes are printed in gray.

In the next scenario, we investigate the impact of the position of a choice-item on the listening behavior of the participants, see Table 7.5. Like in the first scenario, we combine the attribute levels with the lowest impact with the different positions. A choice-item on the first position attracts about 41.4% of the listeners, while the choice-items on position three or two attract only about 29% listeners.

Scenario three, see Table 7.6, shows the impact of the displayed average voting points. Again, we used the attribute levels with the lowest impact and combined them with different levels of the "average voting points" attribute. If zero average voting points are displayed, participants like to listen to this song most, followed by three,

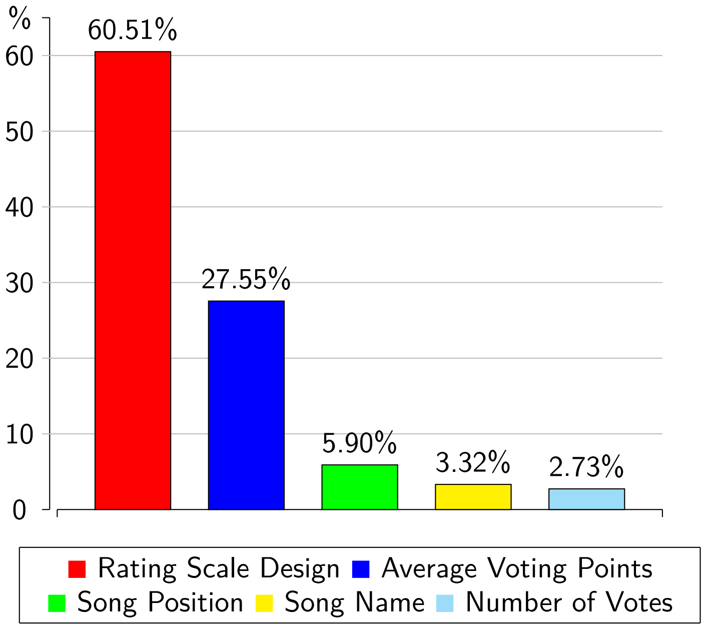


Figure 7.5: Attributes average importance.

four, and five displayed average voting points. If two average voting points are shown, the choice-item animates only 5.3% people to listen to it.

In Table 7.7, twelve scenarios are listed: every choice-set consists of two items. The scenario on the top, Baseline I, consists of the attribute levels with the lowest impact (bad framing): choice-item on position two with thumb rating scale design and a displayed average voting point of five. 50% of the participants would choose the first song to listen to and 50% the second because they have exactly the same attribute levels. In the next eleven scenarios, we change the attribute levels of the second choice-item and compare its attractiveness to the Baseline I scenario. In scenarios four to six, different levels of rating scale design are set. This means, if the participants have the choice to either listen to a song with a thumb rating scale design or one with a heart rating scale design, 61.3% would listen to the song with the heart design–a difference of 11.3% in comparison to Baseline I. With a change from the rating scale design "Thumb" to "Johns", 38.3% of the participants would listen to the latter. If the participants have to choose between a choice-item with thumb design and a choice-set with stars design, 47.2% would prefer to listen to the song with a stars rating scale design.

In comparison to the change of the rating scale design, a different position of the choice-items does not have such a big impact on the choice. A song in position one attracts only 8.4% more listeners in comparison to the other positions. This simulation is presented in the scenarios seven and eight.

Table 7.4: E1: Impact of the rating scale designs

		Framing (Levels of Attributes)			
Scenario	Choice	Position	Design	Av. Voting	Share-MNL
1	1	2	Thumb	5	2.18%
	2	2	Heart	5	3.46%
	3	2	Johns	5	16.84%
	4	2	Stars	5	77.51%

Table 7.5: E1: Impact of the position of the song

		Framing (Levels of Attributes)			
Scenario	Choice	Position	Design	Av. Voting	Share-MNL
2	1	2	Thumb	5	29.33%
	2	3	Thumb	5	29.26%
	3	1	Thumb	5	41.41%

Table 7.6: E1: Impact of the average voting points

		Framing (Levels of Attributes)			
Scenario	Choice	Position	Design	Av. Voting	Share-MNL
3	1	2	Thumb	5	7.70%
	2	2	Thumb	4	10.59%
	3	2	Thumb	3	12.32%
	4	2	Thumb	2	5.33%
	5	2	Thumb	1	25.36%
	6	2	Thumb	0	38.71%

Scenarios 9 to 11 have different levels of the attribute "Average Voting Points". In general, a lower average voting point attracts more participants to listen to a song. This means, in detail, that three average voting points attract 15.8% more listeners, one point 28.6%, and zero 33.5%.

In the last three scenarios, we combine different levels of any attribute together and compare them with the Baseline I scenario. In scenario twelve, we change the song position from position two to position three, we change the rating scale design to the heart design, and we display two average voting points. This has the effect, that 75% of the listeners would choose this attribute level combination; a change of 25%. In scenario thirteen we use the Johns design and show only one average rating point to the participant. Now, 97% would choose this combination of attribute levels; a change about more than 47.2%. Finally, we compare the attribute level combination with the lowest impact (Baseline I) with the attribute level combination with the highest impact: the choice-item is on the first position with stars rating scale design and zero average voting points. This choice-item will attract nearly 99.9% of the listeners and only 0.2% would listen to the choice-item with the bad framing attribute levels.

The result of this study and the simulations are discussed in Section 7.8.

Table 7.7: E1: Impact of design, average voting points, and position in the choice of two songs with bad framing

Scenario	Choice	Framing (Levels of Attributes) Position	Design	Av. Voting	Share-MNL	Change
Baseline I	1	2	Thumb	5	50.00%	
	2	2	Thumb	5	50.00%	
4	1	2	Thumb	5	38.71%	11.29%
	2	2	Heart	5	61.29%	
5	1	2	Thumb	5	11.66%	38.34%
	2	2	Johns	5	88.34%	
6	1	2	Thumb	5	2.81%	47.19%
	2	2	Stars	5	97.19%	
7	1	2	Thumb	5	49.90%	0.10%
	2	3	Thumb	5	50.10%	
8	1	2	Thumb	5	41.57%	8.43%
	2	1	Thumb	5	58.43%	
9	1	2	Thumb	5	34.24%	15.76%
	2	2	Thumb	3	65.76%	
10	1	2	Thumb	5	21.41%	28.59%
	2	2	Thumb	1	78.59%	
11	1	2	Thumb	5	16.51%	33.49%
	2	2	Thumb	0	83.49%	
12	1	2	Thumb	5	24.88%	25.12%
	2	3	Heart	3	75.12%	
13	1	2	Thumb	5	2.76%	47.24%
	2	3	Johns	1	97.24%	
14	1	2	Thumb	5	0.15%	49.85%
	2	1	Stars	0	99.85%	

7.6 Experiment 2 (E2): Exploring the Impact of Service Attributes on the Choice of Rating

7.6.1 The Sample of E2

In total 29 people participated in the experiment 2 (E2). 16 (55%) of them voted for at least one of the tracks, whereas 13 participants (45%) left the experiment without any further action (no-option). Three people (10%) participated three times in the experiment, two people (7%) participated three times, five people (17%) participated two times, and 19 people (66%) once. Most participants, 28 people (97%) could be located in Germany, while accomplishing the experiment. One participant could not be located. The most often used platform for participating in the experiment was Microsoft Windows, in total 21 times (70%). Apple Mac OS was used in six cases (20%) and Linux in three cases (10%). Note that one person opened the experiment with Microsoft Windows and with Apple Mac OS. Regarding the browser, 21 people (70%) used Mozilla's Firefox, four people (13%) used Microsoft's Internet Explorer, four people (13%) used Google's Chrome, and one person (4%) Apple's Safari. Note that one person participated one time by using Firefox and another time by using Chrome.

These data were all extracted from the server logs, the participants were not asked about any demographical data. However, in [TCC07] no impact of gender, age, or education groups by the use of rating systems is measured.

7.6.2 E2 - Step 2: Analysis Results of the Estimation of the Effects and Test of the Significance

Step 2 has been performed for the total data of this study, i.e., of 29 respondents, in total 48 choice-sets. The results of step 2 for experiment 2 are shown in Table 7.8. The LL value is -56.43 for the standard conjoint model, using the estimated effects of Table 7.8. The likelihood ratio test was used to calculate the significance of the standard conjoint model. The chi-square value was 20.22 (degree of freedom $DF = 10$). The standard conjoint model is significant at the 0.05 level.

7.6.3 E2 - Step 3: Analysis Results of the Calculation of the Importance Hierarchy

According to Table 7.9 and Figure 7.6, again the "Rating Scale Design" is the most important attribute (50.81%), followed by "Average Voting Points" (32.03%), and the

Table 7.8: E2: Effect parameters

Attribute Name	Levels	Effect	Std. Error	t-Ratio
Song Name	mf-297	+0.08949	0.26504	+0.33765
	mf-78	+0.21343$^{(*)}$	0.31218	+0.68366
	mf-170	−0.30292*	0.33014	−0.91753
Song Position	1	+0.38574*	0.27261	+1.41498
	2	−0.59666**	0.32572	−1.83184
	3	+0.21091*	0.26473	+0.79670
Rating Scale Design	Stars	−6.03105	–	–
	Heart	−0.97259	91.59698	−0.01062
	Thumb	−1.02209	91.59569	−0.01116
	Johns	+8.02572	91.60277	+0.08761
Number of Votes		−0.29516**	0.13578	−2.17381
Average Voting Points		−1.77212*	0.97926	−1.80965
No Option		+6.29530	91.57381	+0.06875

* Significant at the 0.5 or 0.2 level.
** Significant at the 0.1 or 0.05 level.

"Number of Votes" (11.74%). The attributes representing the song position and the song name have only a minor contribution and influence on the choice of rating audio files.

7.6.4 E2 - Step 4: Simulations of different Choice Tasks

After the analysis of the results, we are now able to perform several simulations, like in Section 7.5.4, to discover the influence of the single attributes in different scenarios. The attributes with the highest contribution levels in the second experiment are: rating scale design, number of votes, and average voting points. Therefore, a badly framed choice-item would consist of the attribute levels with the lowest impact. This means, that the choice-item has the rating scale design "Stars", displays ten votes, and an average voting

Table 7.9: E2: Importance hierarchy and significance level for attribute contribution

IH	Attribute	Attribute importance	Contribution level
1	Rating Scale Design	50.81%	0.0001
2	Average Voting Points	32.03%	0.0001
3	Number of Votes	11.74%	0.0001
4	Song Position	3.55%	minor
5	Song Name	1.87%	minor

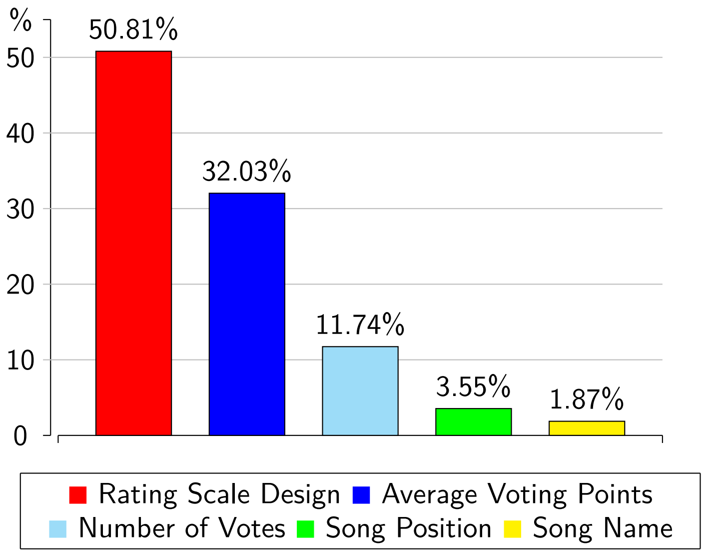


Figure 7.6: Attributes average importance.

point of five. In the tables, gray printed entries indicate a change in the attribute level.

In the first scenario, we want to focus on the impact of the different rating scale designs. To measure the impact of the rating scale design, we display a choice-set with four items to the participant, where every item has a different rating scale design. As a result, nearly all participants, 99.97% (see Table 7.10), vote for a song using "Johns" rating scale design. The other designs attract only few voters.

The next scenario, presented in Table 7.11, illustrates the impact of the attribute "Number of Votes". Again, we use the choice-item with bad framing and change only the displayed number of votes. Starting with ten, the highest number that occurred in our experiment, we decrease the number by two till it reaches zero. A high number of votes distracts participants from voting, therefore, the choice-items one to three with a

Table 7.10: E2: Impact of the rating scale designs

		Framing (Levels of Attributes)			
Scenario	Choice	Design	Number of Votes	Av. Voting	Share-MNL
1	1	Stars	10	5	0.00%
	2	Thumb	10	5	0.01%
	3	Heart	10	5	0.01%
	4	Johns	10	5	99.97%

Table 7.11: E2: Impact of the number of votes

		Framing (Levels of Attributes)			
Scenario	Choice	Design	Number of Votes	Av. Voting	Share-MNL
2	1	Stars	10	5	2.42%
	2	Stars	8	5	4.38%
	3	Stars	6	5	7.78%
	4	Stars	4	5	14.08%
	5	Stars	2	5	25.44%
	6	Stars	0	5	45.90%

high number of votes are chosen only by few participants. But with a smaller number, more participants vote. For this reason, about 46% decided to vote for the choice-item with zero displayed number of votes.

The impact of the attribute "Average Voting Points" is the subject of scenario three, Table 7.12. The results indicate that the participants do not like to vote for a song with many average voting points. Therefore, choice-items with five to two average voting points only attract very few voters. With one average voting point, about 14% of the participants are attracted and with zero average voting points 85%. This emphasizes the awareness of average voting points.

Table 7.13 describes the impact of rating scale design, number of votes, and average voting points in the choice of two songs with bad framing. We compare in scenarios four to six the four different rating scale designs. Here, the weak impact of the stars design becomes obvious: if the design is changed, most participants prefer not to choose

Table 7.12: E2: Impact of the average voting points

		Framing (Levels of Attributes)			
Scenario	Choice	Design	Number of Votes	Av. Voting	Share-MNL
3	1	Stars	10	5	0.02%
	2	Stars	10	4	0.07%
	3	Stars	10	3	0.22%
	4	Stars	10	2	0.70%
	5	Stars	10	1	14.15%
	6	Stars	10	0	84.84%

the stars design to vote for a song. If the Johns design is displayed, all votes are done with this design. Scenarios seven to nine emphasize the impact of the different levels of the attribute "Number of Votes". If two or zero votes are shown, more than 90% of the participants choose one of these choice-items to vote rather than the one with ten shown votes. A change in the levels of the attribute "Average Voting Points" has also a high impact on the choice of the item: a lower number (less than five) will attract all willing voters, zero average votes more than 99.9%. In the last three scenarios, we alternate all three attributes. As a result, no participant would vote for the choice-item with bad framing, all participants (100%) choose different items for voting, independent of the attributes.

Table 7.13: E2: Impact of design, number of votes, and average voting points in the choice of two songs with bad framing

Scenario	Choice	Framing (Levels of Attributes) Design	Number of Votes	Av. Voting	Share-MNL	Change
Baseline I	1	Stars	10	5	50.00%	
	2	Stars	10	5	50.00%	
4	1	Stars	10	5	0.64%	49.36%
	2	Thumb	10	5	99.36%	
5	1	Stars	10	5	0.61%	49.39%
	2	Heart	10	5	99.39%	
6	1	Stars	10	5	0.00%	50.00%
	2	Johns	10	5	100.00%	
7	1	Stars	10	5	23.57%	26.43%
	2	Stars	6	5	76.43%	
8	1	Stars	10	5	8.62%	41.38%
	2	Stars	2	5	91.38%	
9	1	Stars	10	5	5.01%	44.99%
	2	Stars	0	5	94.99%	
10	1	Stars	10	5	2.78%	47.22%
	2	Stars	10	3	97.22%	
11	1	Stars	10	5	0.09%	49.91%
	2	Stars	10	1	99.91%	
12	1	Stars	10	5	0.02%	49.98%
	2	Stars	10	0	99.98%	
13	1	Stars	10	5	0.00%	50.00%
	2	Heart	6	3	100.00%	
14	1	Stars	10	5	0.00%	50.00%
	2	Thumb	2	1	100.00%	
15	1	Stars	10	5	0.00%	50.00%
	2	Johns	0	0	100.00%	

7.7 Experiment 3 (E3): Exploring the Impact of Service Attributes on the Choice of Indicating Interest

7.7.1 The Sample of E3

A sample of 23 people took part in the experiment 3 (E3). 13 of them (66%) visited the product page of at least one track. Ten people (43%) left the experiment without any further action (no-option). Three people (10%) participated four times, two (9%) participated three times, four (17%) participated two times, and 14 (61%) just one time. All of the participants accomplished the experiment out of Germany. 19 people, about 76%, used Microsoft Windows as an operating system to take part in the experiment, four people (16%) used Apple Mac OS, and two people, about 8%, used Linux. Note that two people opened the experiment with Microsoft Windows and Apple Mac OS. 17 people (68%) opened the experiment window with Mozilla Firefox, five (20%) with Google Chrome, two people (8%) with Microsoft Internet Explorer, and one person (4%) with Apple Safari. Note that one person opened the experiment with Internet Explorer and Google Chrome. Another participant opened it with Google Chrome and Apple Safari.

These data were all extracted from the server logs, the participants were not asked about any demographical data. However, in [TCC07] no impact of gender, age, or education groups by the use of rating systems is measured.

7.7.2 E3 - Step 2: Analysis Results of the Estimation of the Effects and Test of the Significance

Step 2 has been performed for the total data of this study, in particular, of 23 respondents, in total 30 choice-sets. The results of step 2 for experiment 3 are shown in Table 7.14. The LL value is -27.39 for the standard conjoint model, using the estimated effects of Table 7.14. The likelihood ratio test was used to calculate the significance of the standard conjoint model. The chi-square value was 28.40 (degree of freedom $DF = 10$). The standard conjoint model is significant at the 0.01 level.

7.7.3 E3 - Step 3: Analysis Results of the Calculation of the Importance Hierarchy

According to Table 7.15 and Figure 7.7, again the "Rating Scale Design" is the most important attribute (42.08%), followed by Average Voting Points (40.42%). The attributes

Table 7.14: E3: Effect parameters

Attribute Name	Levels	Effect	Std. Error	t-Ratio
Song Name	mf-297	−0.47106*	0.57311	−0.82194
	mf-78	+0.58625*	0.69700	+0.84110
	mf-170	−0.11519	0.71506	−0.16109
Song Position	1	+0.67316*	0.41881	+1.60733
	2	−0.03115	0.50965	−0.06112
	3	−0.64201*	0.55846	−1.14961
Rating Scale Design	Stars	−1.71139*	2.89028	−0.59212
	Heart	+3.13757*	3.03834	+1.03266
	Thumb	+2.45224*	3.05235	+0.80339
	Johns	−3.87842*	3.28930	−1.17910
Number of Votes		+0.04943	0.25744	+0.19201
Average Voting Points		+1.34779*	1.10504	+1.21968
No Option		+0.26891	1.14719	+0.23441

* Significant at the 0.5 or 0.2 level.

representing the Song Position, Song Name and the Number of Votes have only a minor contribution and influence on the choice of indicating interest for a specific audio file.

7.7.4 E3 - Step 4: Simulations of different Choice Tasks

With the results from Section 7.7.3 we are able to perform several simulations, like in Section 7.5.4 and 7.6.4, to illustrate the influence of the single attributes in different scenarios. The attributes with the highest contribution levels in the third experiment are: rating scale design and average voting points. Therefore, a badly framed choice-item would consist of the attribute levels with the lowest impact. This means, that the choice-item has the rating scale design "Johns" and an average voting point of zero. In the tables, gray printed entries indicate a change in the attribute level.

In the first scenario (Table 7.16), we simulate a choice-set with four items; every

Table 7.15: E3: Importance hierarchy and significance level for attribute contribution

IH	Attribute	Attribute importance	Contribution level
1	Rating Scale Design	42.08%	0.0001
2	Average Voting Points	40.42%	0.0001
3	Song Position	7.89%	minor
4	Song Name	6.34%	minor
5	Number of Votes	3.26%	minor

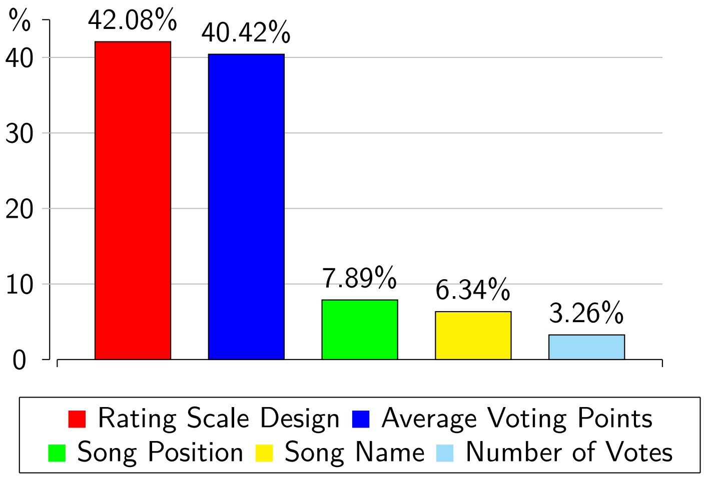


Figure 7.7: Attributes average importance.

item has a different rating scale design. Choice-items with the johns or the stars design do not motivate the participant to visit the product page. When participants visit the product page, they see a choice-item with thumb design (33%) or with heart design (66%).

Listed in Table 7.17, the second scenario investigates the impact of the average voting points on the choice of the participant in visiting the product page of a song. Choice-items with up to three average voting points animate only up to 5% of the participants to visit the product pages of the song. Four average voting points lead 19% of the participants to visit the product page, five average voting points 66%.

The last ten scenarios, presented in Table 7.18, compare two choice-items with the Baseline I scenario. The first choice-item always has a bad framing (like the Baseline I), while the second differs in some of the attributes (different from the Baseline I where

Table 7.16: E3: Impact of the rating scale designs

		Framing (Levels of Attributes)		
Scenario	Choice	Design	Av. Voting	Share-MNL
1	1	Johns	0	0.06%
	2	Stars	0	0.50%
	3	Thumb	0	33.24%
	4	Heart	0	66.16%

Table 7.17: E3: Impact of the average voting points

		Framing (Levels of Attributes)		
Scenario	Choice	Design	Av. Voting	Share-MNL
2	1	Johns	0	0.09%
	2	Johns	1	0.25%
	3	Johns	2	0.38%
	4	Johns	3	4.92%
	5	Johns	4	19.47%
	6	Johns	5	74.88%

the second item also has a bad framing). Scenarios three to five compare the influence of the rating scale designs. If the participants choose between a badly framed song and a song with bad framing, but with a different rating scale design, 90% to 100% would visit the product page of the choice-item with a different rating scale design. This is a change between 40% to 50%. The impact of the average voting points, shown in scenario six to eight, also has a strong influence: if the average voting points are changed from zero to one, 30% of the participants choose this choice-item and visit the associated product page. Three average voting points attract nearly all participants, who are willing to visit the product site. In the last three scenarios, we change both of the attributes, rating scale design and average voting points: to visit the product page, participants would always choose the item with no bad framing.

Table 7.18: E3: Impact of design and average voting points in the choice of two songs with bad framing

Scenario	Choice	Framing (Levels of Attributes) Design	Av. Voting	Share-MNL	Change
Baseline I	1	Johns	0	50.00%	
	2	Johns	0	50.00%	
3	1	Johns	0	10.14%	39.86%
	2	Stars	0	89.86%	
4	1	Johns	0	0.18%	49.82%
	2	Thumb	0	99.82%	
5	1	Johns	0	0.08%	49.92%
	2	Heart	0	99.92%	
6	1	Johns	0	20.42%	29.58%
	2	Johns	1	79.58%	
7	1	Johns	0	1.73%	48.27%
	2	Johns	3	98.27%	
8	1	Johns	0	0.12%	49.88%
	2	Johns	5	99.88%	
9	1	Johns	0	2.04%	47.96%
	2	Stars	1	97.96%	
10	1	Johns	0	0.00%	50.00%
	2	Thumb	3	100.00%	
11	1	Johns	0	0.00%	50.00%
	2	Heart	5	100.00%	

7.8 Discussion

In this chapter, we conducted three experiments to investigate the user's choice of a music song as a cognitive process based on heuristics. In the first experiment, we examined the impact of different attributes on the probability of listening to an audio file. In the second experiment, we observed the impact on the rating behavior of a user. Within the last experiment, we explored the influence of various attributes on the user's choice of buying, i.e., indicating interest in an audio file. In Section 7.2, we introduced a heuristic approach to interpret the user's choice. With the help of the heuristics we are able to explain and to understand the choice of the users. In the next sections, we will discuss the results of each experiment in detail.

7.8.1 E1: Listening to Audio Files

In the first experiment (E1), we examined the influence of the five attributes (Section 7.3.2) in the listening behavior of the visitors. We calculated the importance hierarchy of the attributes with the help of the conjoint analysis, see Section 7.5.3.

Accordingly, the most important attribute in this experiment is the rating scale design. The "Stars" and the "Johns" have a positive effect on the choice of most of the listeners (Table 7.4), which means that these designs attract the listeners. On the contrary, the "Heart" and the "Thumb" design have a negative effect and distract users. An explanation could be the framing effect: the "Stars" and "Johns" designs have a positive effect because of their dominant appearance, i.e., the relatively big size and the flashy colors. The scale design "Stars" is well known from web shops like Amazon and it is likely that users feel attracted to it, which is explained by the availability heuristic. The "Johns" design may trigger positive emotions in the users because of its childlike appearance, explained by the affect heuristic.

The second most important attribute is the "Average Voting Points". Interesting here is that it has a negative effect on the user's choice. This means, if a song has few average voting points, people like to listen to it. The negativity bias could explain this behavior: the negative information encoded in the few "Average Voting Points" can lead them to pay more attention to the song.

The attribute "Song Position" follows the serial position effect. Accordingly, a song in the first position has the highest effect on the user's choice. We observed this primacy effect (or up means good) as well and it has a positive effect on the user's choice, too. The other two song positions have a negative effect on the user's choice. But the last song in the list has a less negative effect than the song positioned in the middle. This is caused by the recency effect. These findings are in line with similar studies that discovered serial position effects (see Section 7.2).

The attributes "Song Name" and " Number of Votes" have only a minor impact on the choice of the participants and can be disregarded.

7.8.2 E2: Rating Audio Files

The next experiment (E2), see Section 7.6, focused on the influence of the five attributes in the rating behavior of the visitors. We discovered that, again, the most important attribute is the rating scale design.

In particular, the "Johns" design has a positive effect on the user's choice. This positive effect could have many reasons: the childlike appearance of this design might cause positive emotions (affect heuristic) and thus attract users for voting. Another reason could be that an unknown scale design makes the users curious and they want to try it out. If rating is seen as an economic process, the funny design of the "Johns" could be interpreted as an incentive for the users and, as such, caused the positive effect. The other designs have a negative effect on the rating behavior, especially the "Stars" design.

The attribute "Average Voting Points" has, like in experiment E1, a negative effect on the user's choice. In particular, a high number of positive votes distract the users from rating while negative votes attract voters. This can be explained with the negativity bias: people pay more attention to negative information. It is also possible that a negative rated song attracts more listeners, see Section 7.8.1, who will vote after listening to a song.

A song with few "Number of Votes" is more attractive for the voters than a song with many votes. An explanation could be that the impact of a single vote is much higher on a song with only few "Number of Votes". Accordingly, the vote of a user has more weight on the rating result of this song. This behavior of the users stands in contrast to the Bandwagon effect (see Section 7.2).

The attributes "Song Position" and "Song Name" have only a minor effect on the user's choice and their contribution can be disregarded.

However, this experiment has some limitations: as listed in Table 7.9 the contribution levels of rating scale designs is very high, but as seen in Table 7.8 and Table 7.10 the effect of the single rating scale design is not significant. Further studies with more participants are necessary to confirm or discard our findings.

7.8.3 E3: Indicating Interest

The last experiment (E3) has only two attributes with a major impact on the choice of the users. If visitors indicate their interest to buy a song by clicking on the link to the product site, the “Rating Scale Design” has the biggest importance. In detail, the “Heart” and the “Thumb” designs have a positive impact, while the “Stars” and the “Johns” designs have a negative impact. In comparison, the design “Heart” and the design “Thumb” are smaller than the design “Stars” and the design “Johns” so that the link to the product page becomes more visible and more users click on it (framing effect).

The “Average Voting points” have a positive effect on the user’s choice: the better a song is rated, the more visitors click on the link to the product page of the song. It is not surprising that the customers want to buy a well rated song and not a badly rated one. This positivity bias is observed in hedonic products, as described in Section 7.2.

The other attributes have only a minor effect on the user’s choice and can be disregarded.

7.9 Conclusion and Future Work

The main finding, summarized in Table 7.19, of the three experiments is that the rating scale design always has a high effect on the user’s choice. A web shop owner who wants to attract many voters or listeners should consider adding new, uncommon, and eye-catching scale designs to draw the attention to the rating system. Web shop owners who want to increase their sales should use more discreet scale designs, so that the customers do not overlook the other part of the shopping page, i.e., the buy button.

If a web shop owner decides to use a scale where the average voting points are displayed, she/he should remember the negative effect on the rating and the listening behavior of the users. But a hedonic product with good votes may be more interesting to customers who are interested in buying.

The song position has an effect on the user’s choice only when the user listening to an audio file. The results of our experiments support former researches in that area: the song in the first and last position attracts the most listeners while the song in the middle is not so attractive for the users.

Web shops that are interested in motivating their customers to rate their products should include also the total number of votes in their rating scale, but maybe only to those products that have few votes. That could increase the user’s motivation to rate these products as well.

The song name has no effect in our experiments. We chose very uncommon names on purpose so that the customers are not attracted by famous song names. Our strategy was successful but it may be interesting to try a new experiment to compare the effect of common and uncommon song names on the user’s choice.

The new scale design “Johns” was very popular in two of the three experiments, so

Table 7.19: Main findings

Attribute Name	Hierarchy Level / Effect	Heuristic / Finding
Song Name	only minor effect (E1, E2, E3)	-
Song Position	the song on the first position has the highest impact, followed by the last positioned song in the list (E1)	serial position effect, primacy effect, recency effect
Rating Scale Design	high effect in all experiments (E1, E2, E3)	affect heuristic, availability heuristic, framing effect, incentive
Number of Votes	negative effect on rating a song (E2)	impact of a single vote
Average Voting Points	negative effect on listen to a song (E1) and rate a song (E2); positive effect on indicating interest in a song (E3)	negativity bias, positivity bias, bandwagon effect

that an experiment with more new scale designs, i.e., maybe more personalized designs, would be interesting.

It would also be interesting to perform an experiment on how users interpret the different levels of unlabeled rating scales, e.g., of the "Stars" design. On almost every web shop, the rating scale levels are not explained to the users, so that customers could interpret the levels differently: for example, one customer could express himself by rating a product with only one star that she/he dislikes this product, while another customer expresses with one star that she/he likes the product, but only a little.

Table 7.20: Contribution summary

Subject	***Detail***
Cognitive Process of Rating	We adapted the theoretical framework of the cognitive response process from Tourangeau et al. [TRR00] to the cognitive process of rating (Section 7.2)
Influencing Factors on User Feedback	We conducted three experiments to discover the influencing factors on user feedback: E1: Impact of service attributes on the choice of listening to audio files (Section 7.5) E2: Impact of service attributes on the choice of rating (Section 7.6) E3: Impact of service attributes on the choice of indicating interest (Section 7.7)
	We evaluated the three experiments with the help of a choice-based conjoint analysis in four steps: Step 1: choice and definition of the estimation model (Section 7.4.1) Step 2: estimation of the effects using the estimation model and test of the significance of the effects and of the standard conjoint model (Section 7.4.2, 7.5.2, 7.6.2, and 7.7.2) Step 3: calculation of the importance hierarchy and the contribution size of the attributes (Section 7.5.3, 7.6.3, and 7.7.3) Step 4: simulations of different choice tasks (Section 7.5.4, 7.6.4, and 7.7.4)
	We interpreted the results of the three experiments according to the cognitive process of rating and proposed guidelines for web site owners on how to place and enrich rating systems on web pages.

Conclusion

8

Conclusion and Future Research Directions

8.1 Conclusion

This thesis provides a conceptual framework and an implementation of a system that helps to *better* understand the *behavior* and potential *interests* of web site visitors by keeping into account, both, the *explicit* and *implicit feedback*. This thesis is divided into two parts.

The first part, rooted in computer science and information systems, is using graph theory and an extended click-stream analysis in order to define a framework and a system tool that is useful for analyzing web user behavior by calculating their *interests*.

The second part, rooted in behavioral economics, mathematics, and psychology is investigating *influencing factors* on different types of web user's choices. In detail, a model for the *cognitive process* of rating products on the Web is defined and an *importance hierarchy* of these influencing factors is discovered.

We used an interdisciplinary approach to address two main research questions.

Q1: What do we learn from analyzing explicit and implicit user feedback on the Web?

The thesis contributions in this research area are the following: To better understand web user behavior, we developed in Chapter 3 *a framework analysis for managing the feedback of visitors of a web site*. Within seven steps, this framework analysis helps web site owners to check the *consistency* of the behavior of the web site visitors by analyzing a variety of feedback obtained from them. We defined a new concept for a user profile, called the *Relevance Profile*, and four consistency levels. With the help of three case studies, we demonstrated in Chapter 5 the applicability of these new concepts to helping to discover *patterns of usage*, such as indicators for structural problems of a web site, topic term misunderstandings, lost interest in a topic, and stable interest in a topic.

In Chapter 4, we presented a *tool for building and mining similarity graphs*. This tool provides the web site owner with an overview of the web community with respect to the *similarity* and *importance* of the users' interests. We presented four different alternatives to building a similarity graph. In addition, we provided nine algorithms, including two that were original contributions of this thesis, for calculating the importance of the users in the community. We demonstrated the applicability of this tool by conducting three case studies in Chapter 5.

In Chapter 6, we enhanced the *accuracy* of the user profiles, collecting more behavioral data on the users by using a new source that is different from the server log files. For this reason, we extended the Gugubarra Framework with the capability to track and analyze the *mouse activities* of the web site visitors. We integrated mouse-tracking into the existing concepts of the Gugubarra Framework and evaluated mouse-tracking ability with an experiment, showing that the user profiles are now more accurate.

Q2: What are the main factors that influence user feedback on the Web?

The thesis contributions in this research area are the following: In Chapter 7, we extended the Gugubarra Framework, adding guidelines on *how to place and enrich rating systems* on web pages. We investigated the *influence factors* of various rating scale designs on different types of user's choices. Hence, we defined a model for the *cognitive process* of rating audio files on the Web. Within the four components of this model, we presented *cognitive heuristics* that can be placed in context for judging and rating songs. Furthermore, we conducted three experiments and identified an *importance hierarchy* of the influencing factors on the user's choices with the help of a choice-based conjoint analysis. More specifically, in the experiments, we explored the impact of service attributes on the choice of listening to, rating, and indicating interest in audio files.

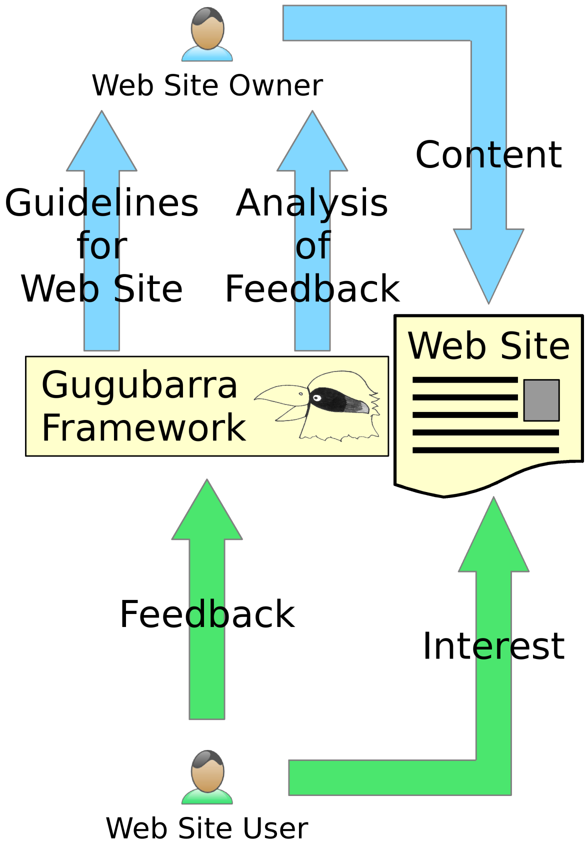


Figure 8.1: Functional elements of the Gugubarra project

In summary: With the research results of this thesis, a web site owner has the opportunity to analyze the feedback of a *single* user as well as of an *entire* web community, including the mouse activities of the web site visitors. In addition, this thesis explored new guidelines on *how to place and enrich rating systems on web pages*. With this knowledge about the interests of the visitors, the web site owner can adapt web pages accordingly and provide proper content. With guidelines on how to collect feedback on the visitors, the web site owner can collect suitable feedback for the analysis. Figure 8.1 illustrates the functional elements in the context of the Gugubarra project.

8.2 Future Research Directions

This thesis contributes to many different areas of web user behavioral analysis and proposes some possible future research directions, which are briefly explained in the following:

Exploring new web technologies to infer web user behavior: The current version of the Gugubarra Framework is a *web site owner centric* system, which means that the web site owner has full control over the system and the web site users have none. In a future version, it could become a more *user centric system* by using the collective intelligence of the World Wide Web community. For example, *tags*[1] could be used to involve web site visitors in the process of assigning topics to zones. Even the weights of the user topics could be determined by simply counting how often the same tag word was added to a zone. With tagging [Jaz07], we would also gain a new source of explicit user feedback. However, this approach would imply a loss of control of the web site owner over her/his web site.

The new *HTML5*[2] standard promises to introduce a multitude of exciting new features, which could be used to further improve the accuracy of our tracking-based mouse profiles. Especially the upcoming *user attributes* feature appears to be convenient for achieving such improvements.

Improving the web analytic usability of Gugubarra: In the future, the *set of consistencies* could be reduced to improve the usability of the consistency check. The results of the three case studies indicate that some consistency levels have the *same expressiveness*. This task could be solved by performing more case studies, as we did in Chapter 5. By performing more case studies, more indicators for weak/strong points of a web site could be detected as well.

Aside from the extension of the tool for building and mining similarity graphs with more algorithms for the similarity calculations, the exploration of *similarity (or importance) metrics* would be reasonable. With this type of metric, an *objective* evaluation of the similarity algorithms would be possible, which would help web site owners to choose a *suitable* algorithm for the similarity calculation.

Inferring mobile web user feedback: With the ability to track the mouse activities of the web site visitors of the Gugubarra Framework, we could explore the behavior of users of new device classes, such as *smartphones* or *tablet computers*. These devices are usually not controlled via a regular mouse device but instead are controlled via a touchscreen. This new input technique could affect the behavior patterns of the web users and requires rethinking some of the concepts of web user behavioral analysis.

[1]Tags are meaningful keywords that are added by the users of a web resource. These words can describe the content of that resource, e.g., an image or a Gugubarra zone. The result is a flat taxonomy [GH06] for the tagged resource built by the users (the folk); therefore, tags are sometimes called folksonomy [VW07].

[2]http://dev.w3.org/html5/spec/

Even the analysis of the navigation data of the GPS[3], which is built into the most new portable devices, could be an interesting source of user feedback.

Follow-up experiments on exploring influencing factors on web user feedback: During the three experiments on exploring the impact of service attributes on the user's choice, the attribute "Song Name" had no effect because we chose very uncommon names on purpose to avoid having the customers be attracted by famous song names. Our strategy was successful but it may be interesting to attempt a new experiment to *compare the effect of common and uncommon song names on the user's choice*.

The new scale design "Johns" was very popular in two of the three experiments, so that an experiment with more new scale designs, i.e., perhaps more *personalized designs*, would be interesting.

It would also be interesting to perform an experiment on *how users interpret the different levels of unlabeled rating scales*, e.g., of the "Stars" design. In almost every web store, the rating scale levels are not explained to the users; as a result, customers could interpret the levels differently: for example, one customer could express himself by rating a product with only one star when she/he dislikes the product, while another customer could express with one star that she/he likes the product, but only a little.

Data privacy: One future perspective, which should not be forgotten, is the ethical issue of data privacy, which is related to the use of behavioral analysis technologies. Proper and responsible use of the data must always be ensured.

[3]Global Positioning System

Appendix

A
Appendix

A.1 Case Studies: Explicit Feedback

A.1.1 Invitation

```
Dear participant,
please read the following test sequence exactly. Afterwards, you can
go step by step thru the test:

Step 1. Please give us an initial feedback about
        your interests: <click here>

Step 2. Please select and execute at least one task
        from the list below:

Teaching:
* In which term we conduct the DB2 lesson?
* On which day the fifth exercise sheet was discussed in the
  summer term 2008?
* How many exercises were already held in this term?
* What is the title of the diploma thesis of the student Florian
  Quetting?
```

Research:
* How many research projects were conducted by the DBIS group?
* When did Prof. Roberto V. Zicari a talk about the Gugubarra project at the Google headquarter in the U.S.?
* Who was the contact person at DBIS of the ABILITIES Project?
* How many internal PhDs are currently at the DBIS group?

News:
* How many news items are published on the ICOODB 2010 conference and who are the keynote speakers of this conference?
* Who was guest-lecturer on 3. February and where does this person live?
* In which hall and which stand at the CeBIT 2006 the DBIS group presented the Gugubarra project?
* How many news items are published on database systems?

Step 3. Please give now a final feedback about
your interests: <click here>

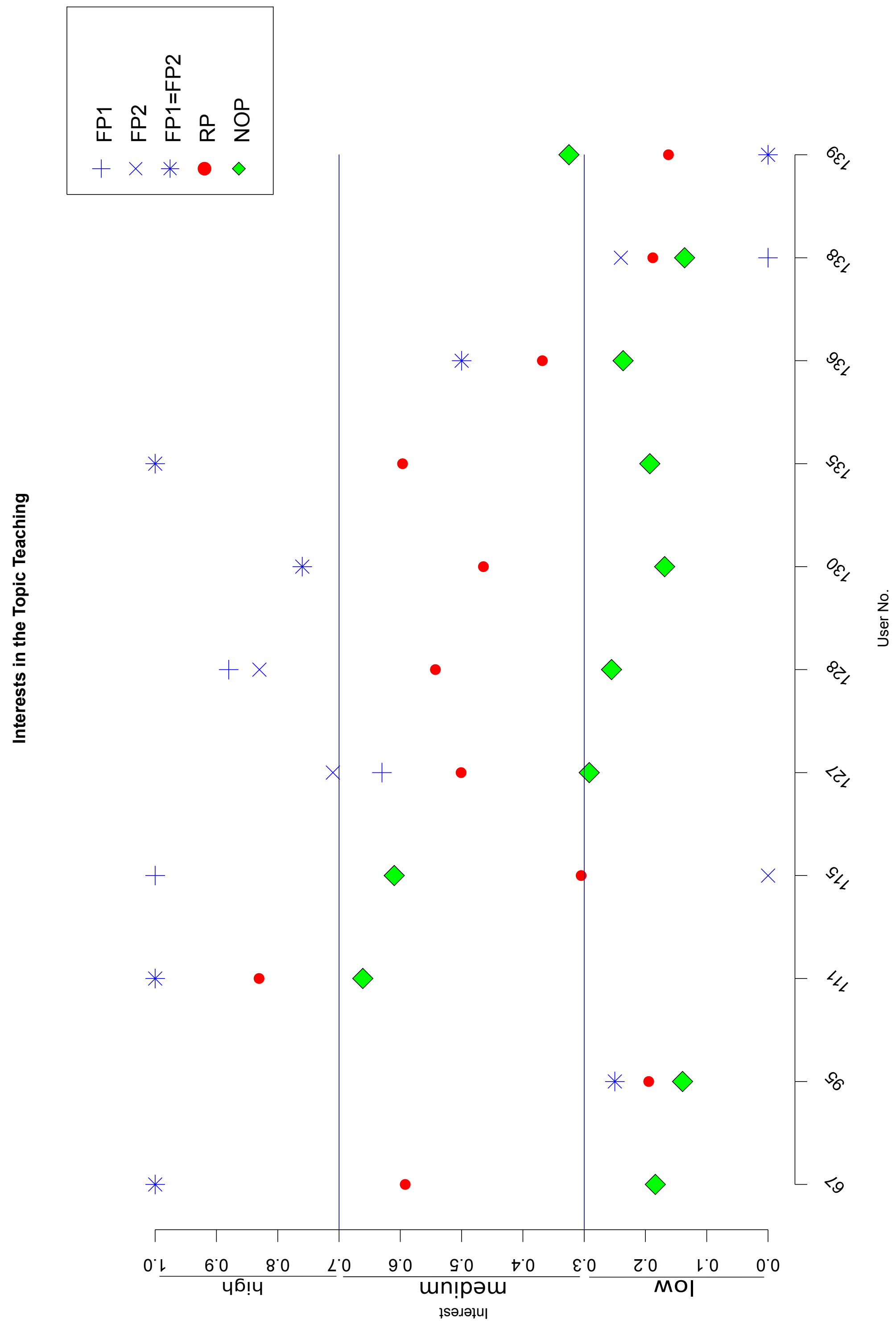


Figure A.1: Feedback case study: NOP, RP, and both FPs in the "teaching" topic

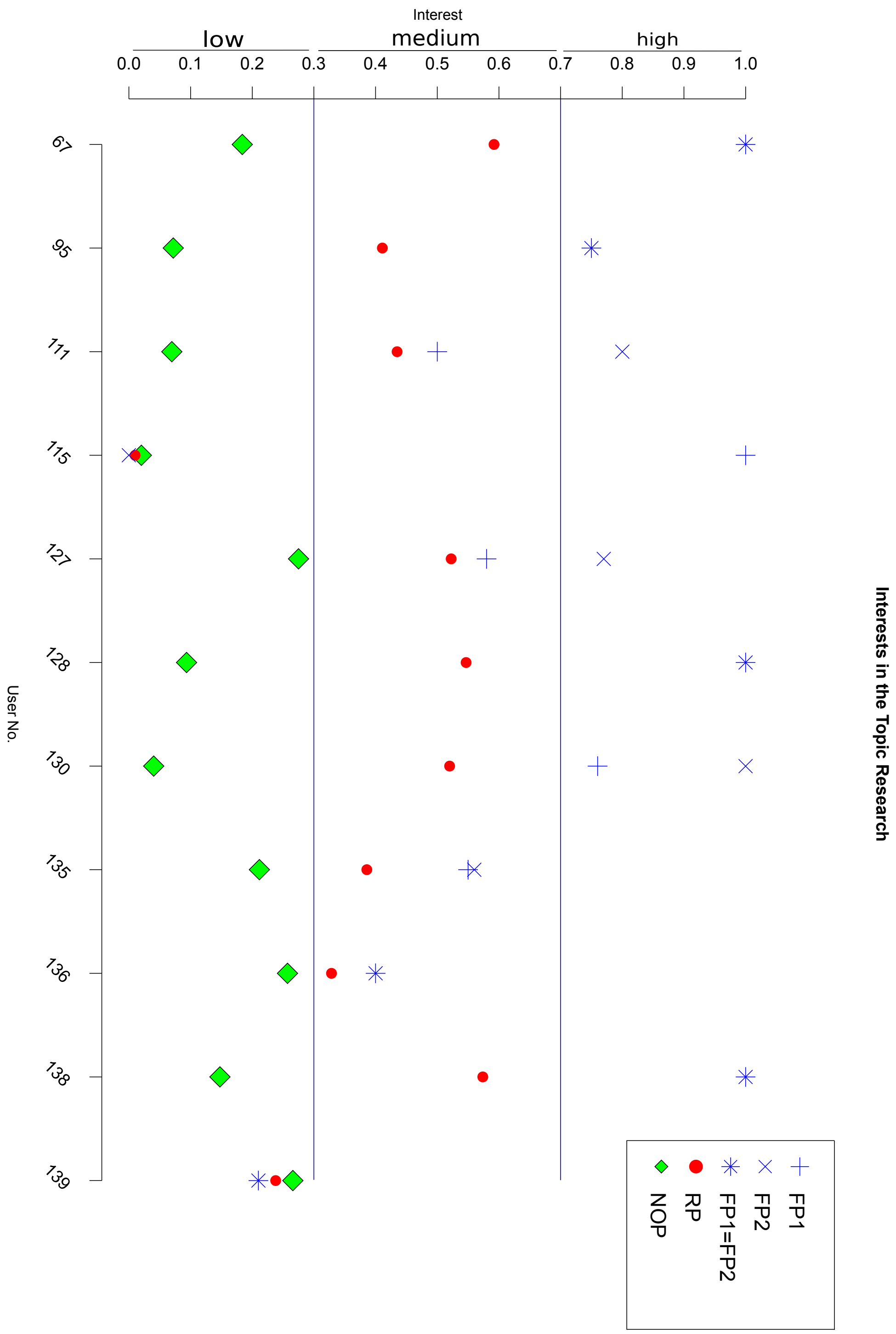


Figure A.2: Feedback case study: NOP, RP, and both FPs in the “research” topic

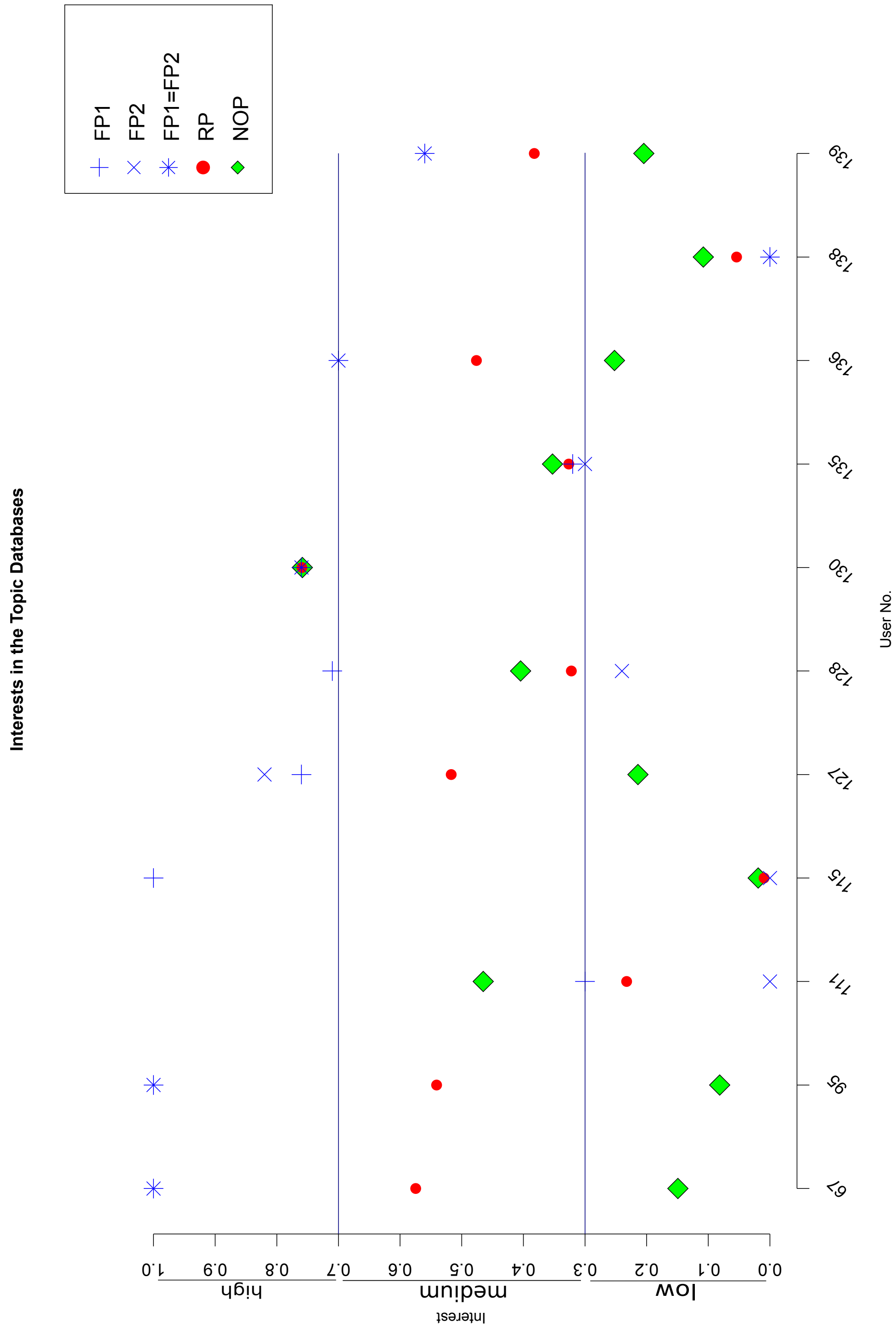


Figure A.3: Feedback case study: NOP, RP, and both FPs in the “databases” topic

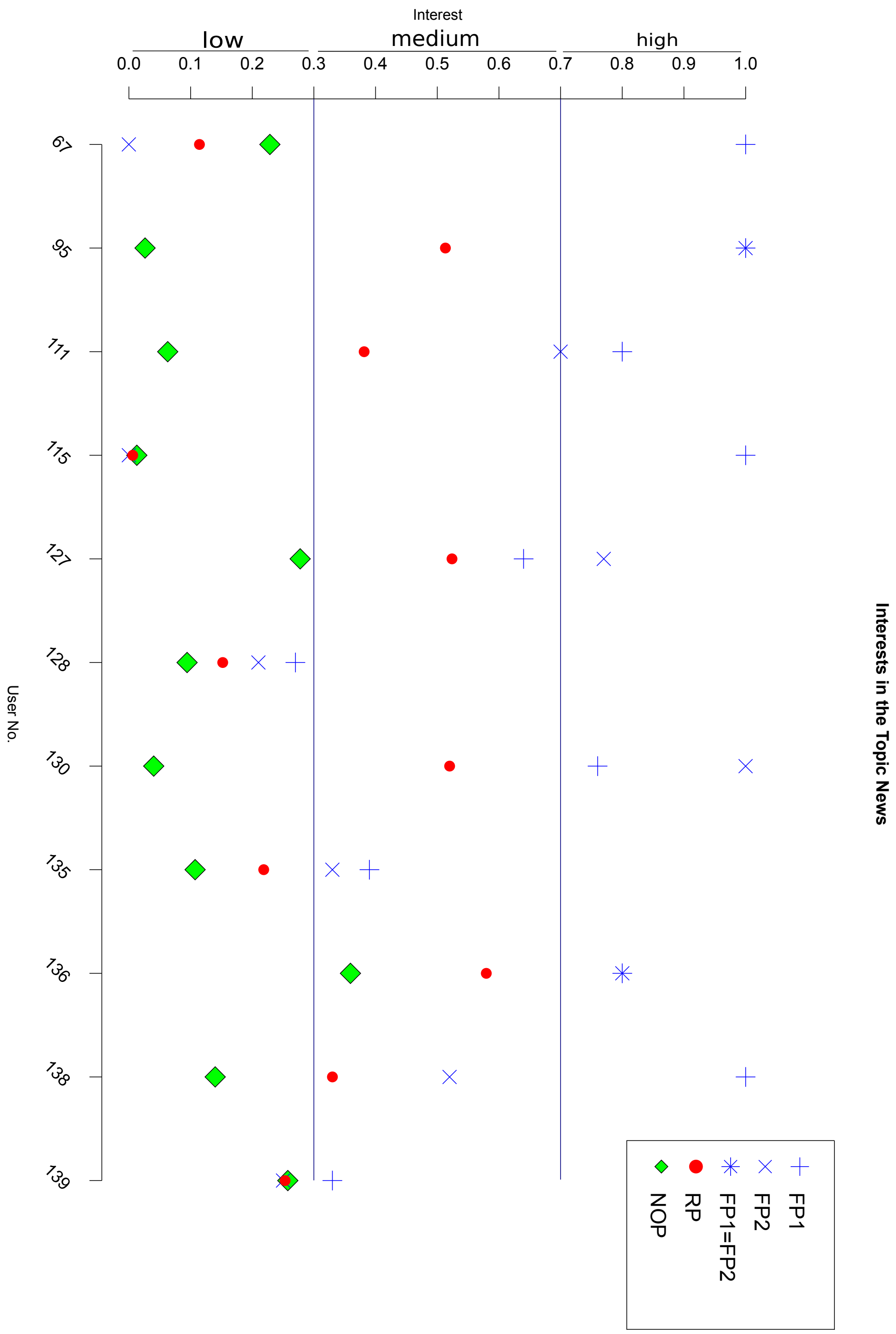


Figure A.4: Feedback case study: NOP, RP, and both FPs in the "news" topic

A.2 Case Studies: Cold-Warm-Start

A.2.1 Invitation

```
Hallo Versuchsteilnehmer/in,

vielen Dank für Ihre Teilnahme an meinem kurzen Experiment
zur Verifizierung unserer Web Analytics Software Gugubarra.
Das Experiment beginnt auf einer Web Seite mit einer Liste
von Links zu unterschiedlichen Themenbereichen. Klicken Sie
einfach auf den Link zu dem Thema, dass Sie am meisten
interessiert und geben Sie uns zum Abschluss noch kurz Ihr
Feedback zu Ihren Interessen.
Nach einer Woche werden wir das Experiment wiederholen, um
unsere Daten zu überprüfen. Sie werden dann noch einmal eine
E-Mail von mir erhalten.
Bitte klicken Sie auf den unten stehenden Link und loggen Sie
sich mit folgenden Daten ein, um das Experiment zu starten:

Benutzername: xxxxxx
Passwort: xxxxxxxx

http://www.dbis.cs.uni-frankfurt.de/index.php/gugubarra-ver...


Vielen Dank für Ihre Hilfe,
Clemens Schefels

Goethe-Universität Frankfurt
Professur Datenbanken und Informationssystem (DBIS)
Robert-Mayer-Str. 10
60325 Frankfurt am Main
Tel: +49 0(69) 798 22426
Fax: +49 0(69) 798 25123
```

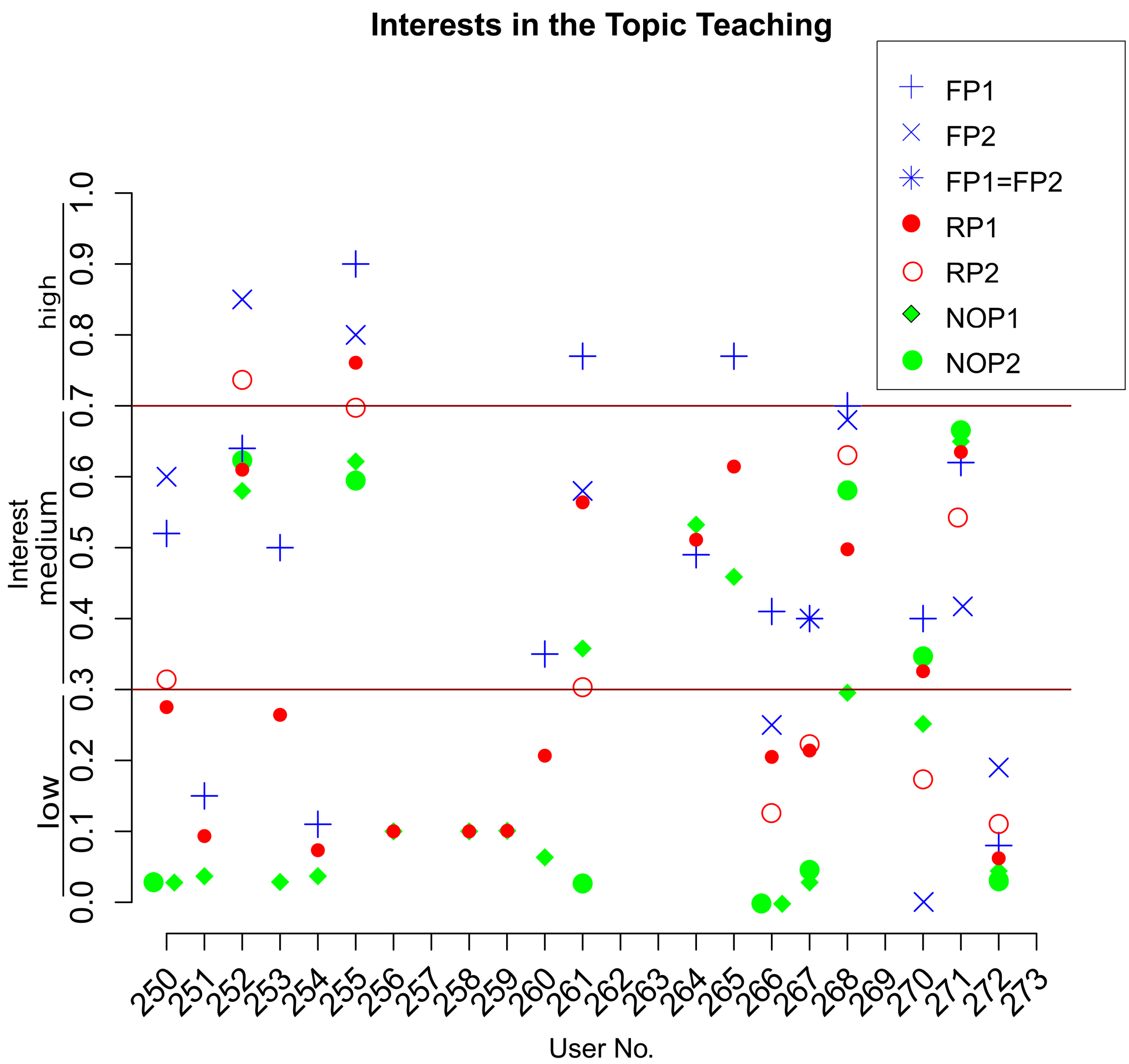


Figure A.5: Cold-warm-start case study: NOPs, RPs, and FPs in the “teaching” topic

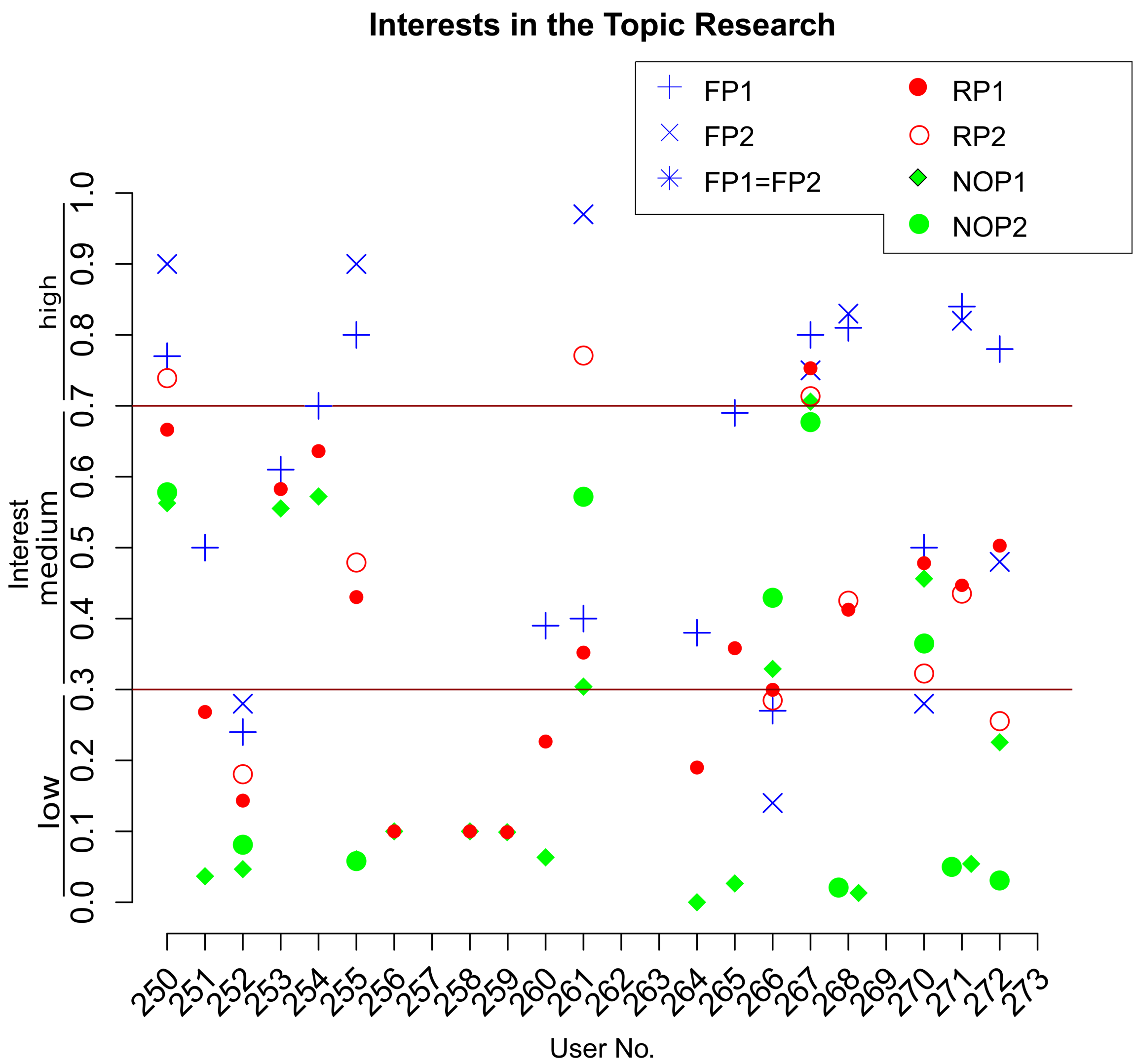


Figure A.6: Cold-warm-start case study: NOP, RP, and both FPs in the "research" topic

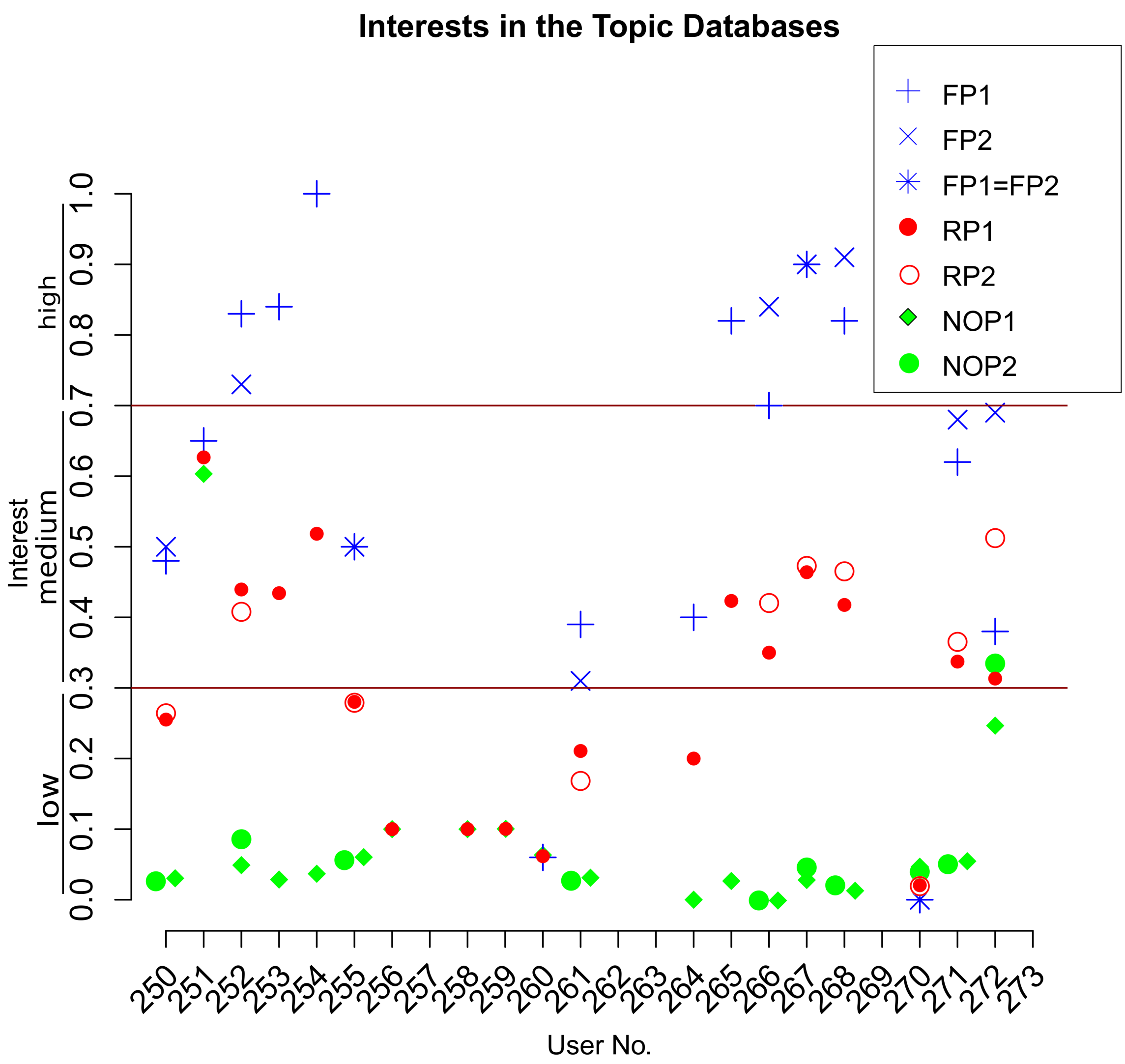


Figure A.7: Cold-warm-start case study: NOP, RP, and both FPs in the “databases” topic

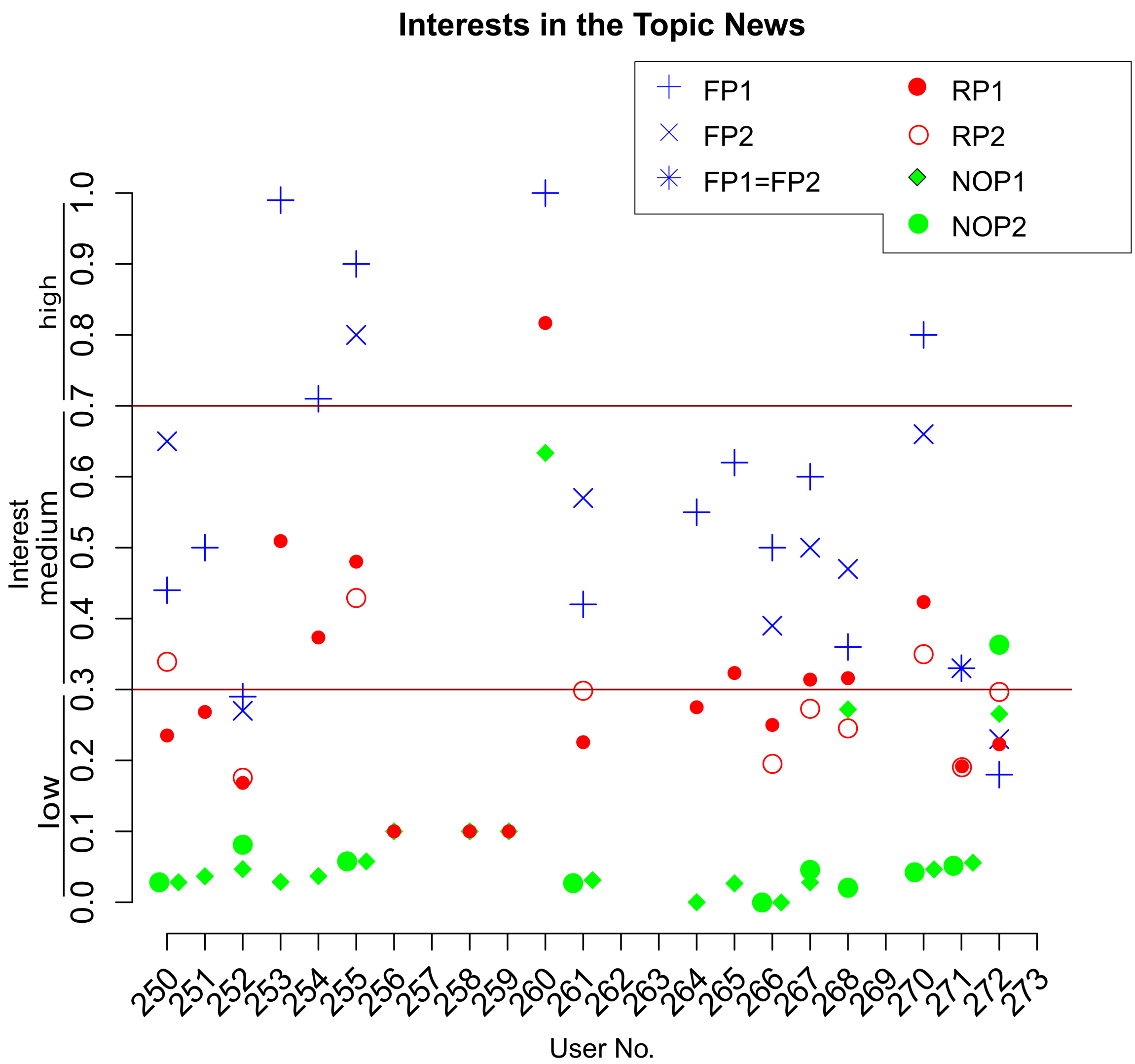


Figure A.8: Cold-warm-start case study: NOP, RP, and both FPs in the "news" topic

A.3 Case Studies: All Data of Gugubarra

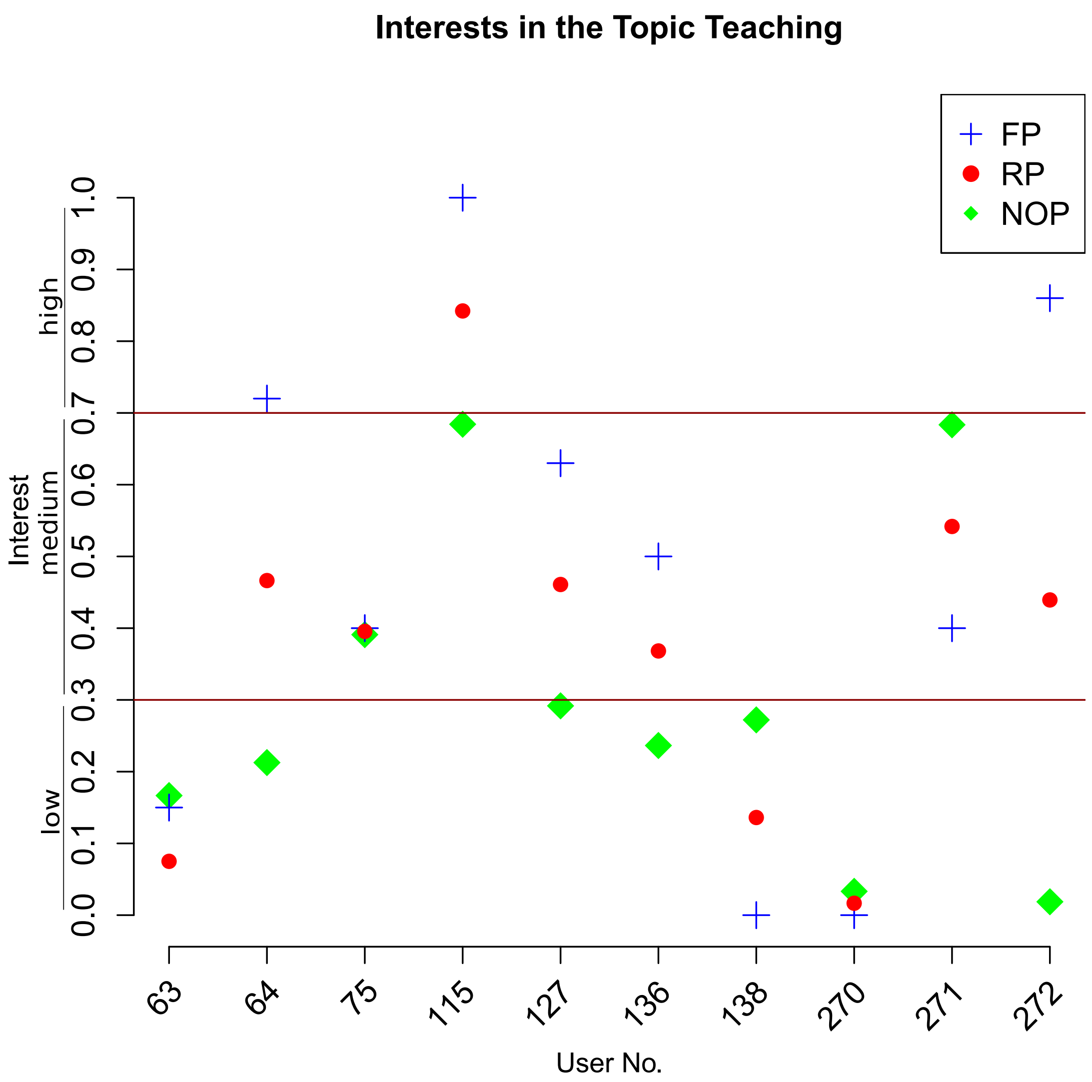


Figure A.9: All users case study: NOP, RP, and both FPs in the "teaching" topic

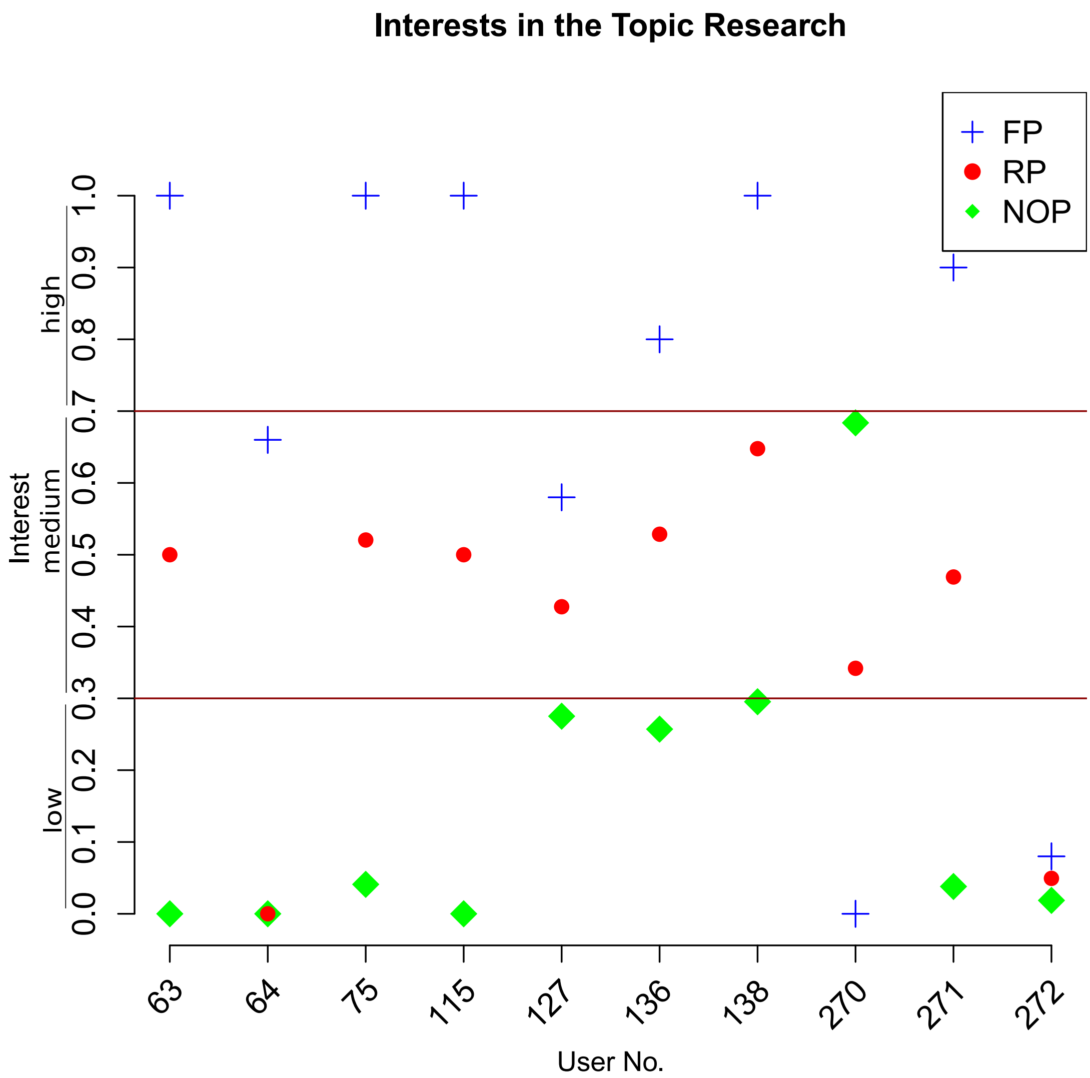


Figure A.10: All users case study: NOP, RP, and both FPs in the “research” topic

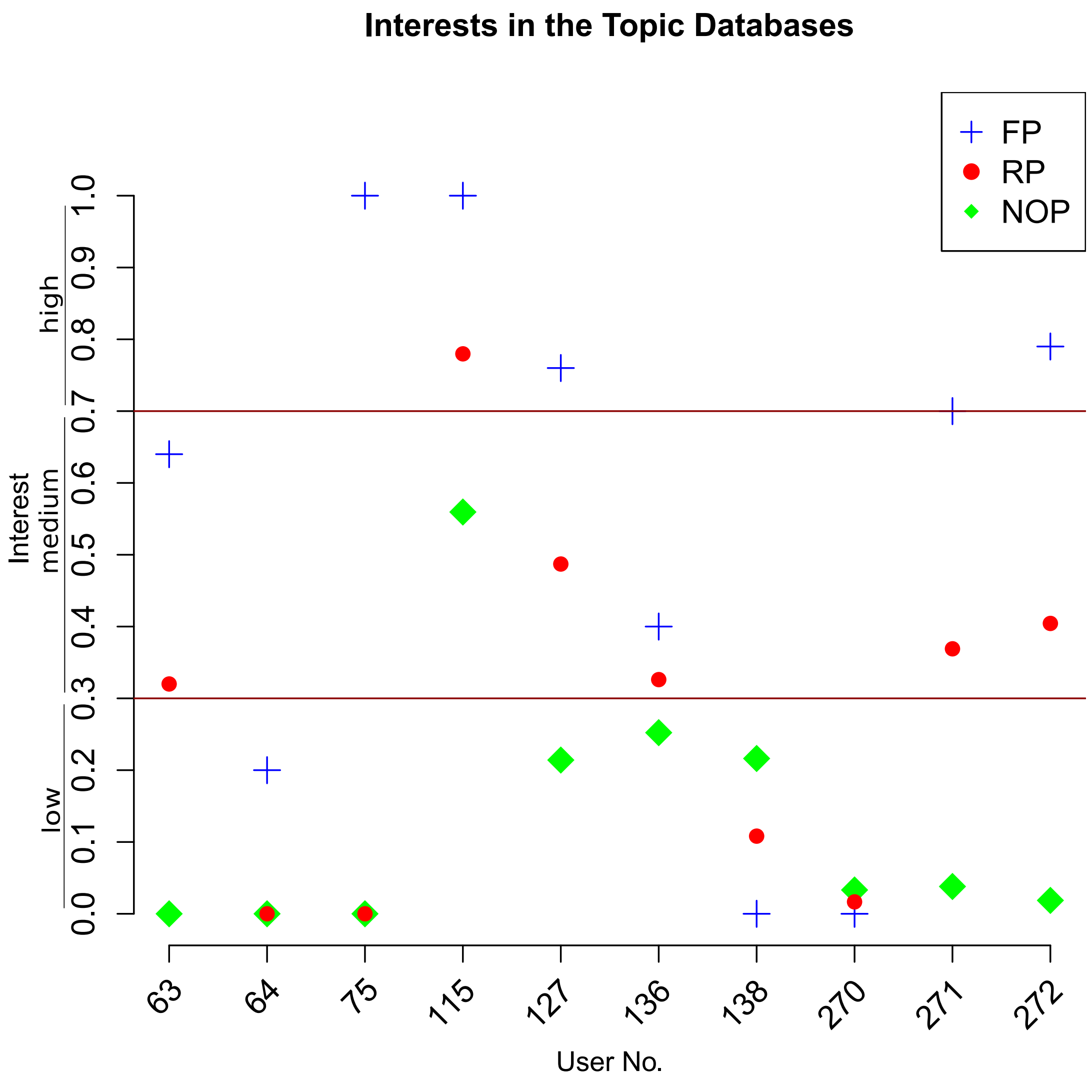


Figure A.11: All users case study: NOP, RP, and both FPs in the "databases" topic

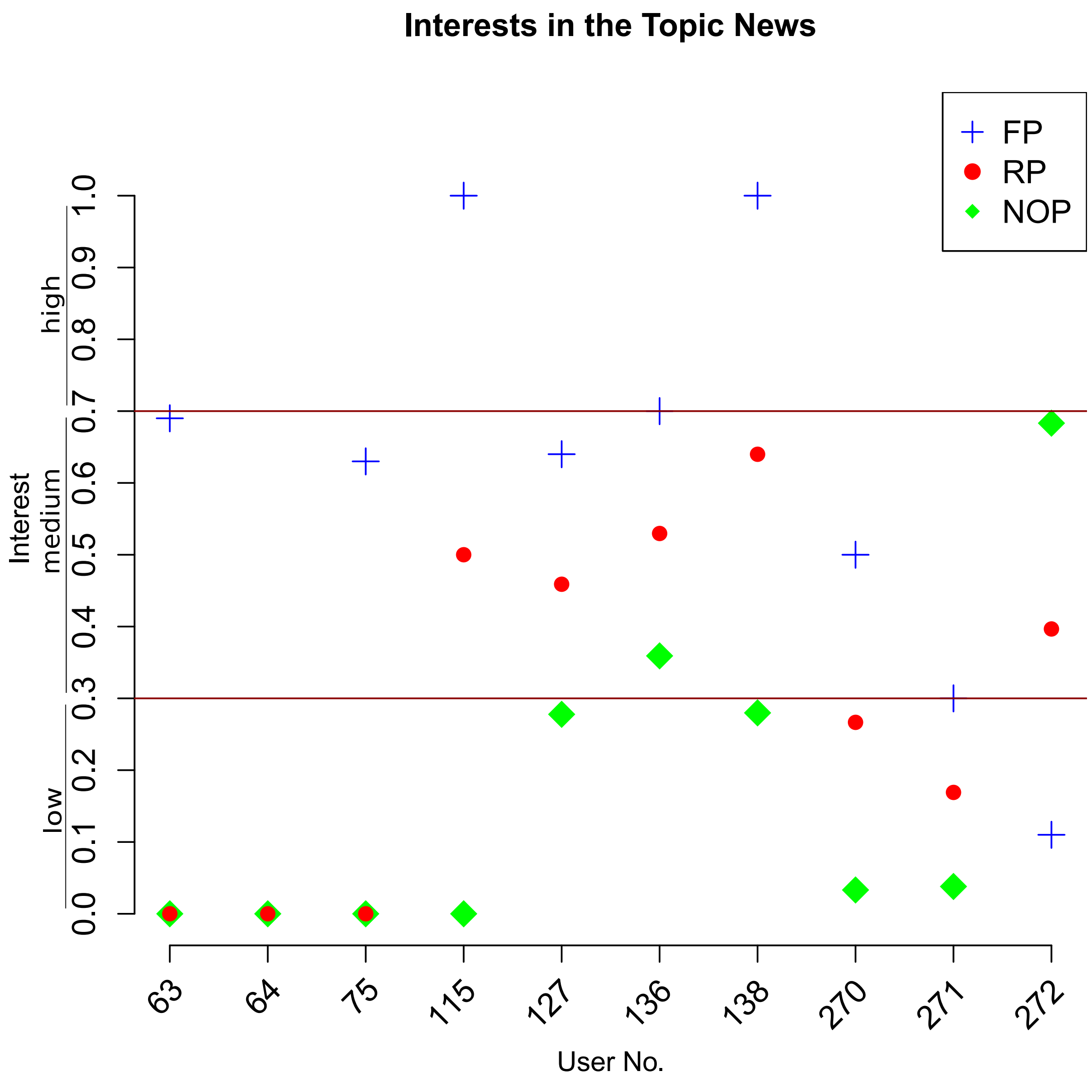


Figure A.12: All users case study: NOP, RP, and both FPs in the “news” topic

Nomenclature

δ_{i_f}	durations for page visits (see duration)
$C^{e\downarrow}$	down explicit consistent
$C^{i\downarrow}$	down implicit consistent
C^{e}	explicit consistent
C^{i}	implicit consistent
C^{t}	total consistent
$\not{C}^{t}$	total inconsistent
$C^{e\uparrow}$	up explicit consistent
$C^{i\uparrow}$	up implicit consistent
A	adjacency matrix
$ActP$	Action Profile (part of NOP)
$aspl$	average shortest path length
$aspw$	average shortest path weight
aw	action weight
C_r	Range Centrality
c_s	single mouse click
c_{db}	double mouse click
cat_x	ordinal category

CIP	Click Profile (part of MP)
$Cluster_{T_i}$	cluster of topics
$Cluster_{u_m}$	cluster of users
D	ordered set of ordinal categories
$duration$	time a user spent on a web page
$DurP$	Duration Profile (part of NOP)
$E(G)$	set of edges of the graph G
e_i	mouse-event no. i
F	location variable
$F^{\uparrow}$	location variable
f_l	filter function: defines the relative importance of a user feedback
FP	Feedback Profile (explicit feedback)
G	graph
L_1-norm	Manhattan Distance
MP	Mouse Profile (implicit feedback)
N	location variable
$N^{\uparrow}$	location variable
NOP	Non-Obvious Profile (implicit feedback)
OP	Obvious Profile (e.g., name, age, address)
P	web page
R	location variable
RP	Relevance Profile (container for all feedback profiles)
S_k	scope of the filter function f_l

T_i	mouse trail (in context of mouse-tracking)
T_i	topic no. i
t_n	point in time no. n
u_m	user no. m
v	topic weight
$V(G)$	set of vertices of the graph G
Z	Gugubarra zone
ZvP	Zone visiting Profile (part of MP)
AJA-JSON	asynchronous JavaScript and JSON
AJAX	asynchronous JavaScript and XML
API	application programming interface
CBC	choice-based conjoint analysis
CG	complete graph
CMS	content management system
CNG	closest neighbor graph
CS	choice-set
CT	choice task
DBIS	Databases and Information Systems research group
DF	degree of freedom
DOM	document object model
e-WOM	electronic word-of-mouth propaganda
E1	experiment 1: exploring the impact of service attributes on the choice of listening to audio files

E2	experiment 2: exploring the impact of service attributes on the choice of rating
E3	experiment 3: exploring the impact of service attributes on the choice of indicating interest
FIFO	first in, first out
GPS	Global Positioning System
Gugubarra	Gugubarra Framework
Gugubarra Analyzer	part of the Gugubarra Framework
Gugubarra Designer	part of the Gugubarra Framework
Guguboomla	Gugubarra Designer Joomla! plug-in
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
igraph	R library
IH	attribute importance hierarchy
IP-address	Internet protocol address
JIT compilation	just in time compilation
JS DOM Event Object	JavaScript DOM Event Object
JSON	JavaScript object notation
LL	log-likelihood
location	ordinal category
MNL	multinomial logit model
MST	minimum spanning tree
on-line community	all registered users of a web site

R	statistic programming language
S	statistic programming language
SAM	self-assessment manikin
SCG	smallest connection graph
Share-MNL	share of preference model using MNL
Std. Error	standard error
user session	time between the log-in and the log-out of a web user
web community	all registered users of a web site
web site	collection of web pages
web site manager	controls the contents of a web site
web site owner	controls the contents of a web site and decides on the business strategies or goals
web site user	visitor of a web site
web site visitor	visitor of a web site
WOM	word-of-mouth propaganda
XML	Extensible Markup Language

List of Figures

List of Tables

Index

Bibliography

[AA03] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003. 50

[ABD06] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–26, New York, USA, 2006. ACM. 39

[AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory (ICDT '01)*, pages 420–434, Berlin, Germany, 2001. Springer. 45

[AS07] Richard Atterer and Albrecht Schmidt. Tracking the interaction of users with ajax applications for usability testing. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1347–1350, New York, USA, 2007. ACM. 100

[ASW06] Ernesto Arroyo, Ted Selker, and Willy Wei. Usability tool for analysis of web designs using mouse tracks. In *CHI '06: CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pages 484–489, New York, USA, 2006. ACM. 100

[AWS06] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 203–212, New York, USA, 2006. ACM. 100

[Bau92] Hans Baumgartner. Remembrance of things past: Music, autobiographical memory, and emotion. *Advances in Consumer Research*, 19(1):613–620, January 1992. 125

[BBPSV04] A. Barrat, M. Barthélémy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004. 48

[BEPW05] K. Backhaus, B. Erichson, W. Plinke, and R. Weiber. *Multivariate Analysemethoden: Eine Anwendungsorientierte Einführung*. Springer-Lehrbuch. Springer, Berlin, Germnay, 2005. 131, 132

[BFJ+12] Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam D.I. Kramer, Cameron Marlow, Jaime E. Settle, and James H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, September 2012. 126

[BL94] Margaret M. Bradley and Peter J. Lang. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59, 1994. 127, 129

[BLHH+06] Tim Berners-Lee, Wendy Hall, James Hendler, Nigel Shadbolt, and Danny Weitzner. Creating a science of the web. *Science*, 313(5788):769–771, August 2006. VII, 4

[Bol98] Béla Bollobás. *Modern Graph Theory*. Graduate Texts in Mathematics. Springer, Berlin, Germany, 1998. 46, 47

[Bon87] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, March 1987. 48

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, volume 30, pages 107–117, Amsterdam, The Netherlands, 1998. Elsevier. 48

[CAS01] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. What can a mouse cursor tell us more?: Correlation of eye/mouse movements on web browsing. In *CHI '01: CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 281–282, New York, USA, 2001. ACM. 100

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance / similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007. 45

[CHZ11] Ioanna Constantiou, Natascha Hoebel, and Roberto V. Zicari. How do framing strategies influence the user's choice of content on the web? *Concurrency and Computation: Practice and Experience*, 22(7):1794–1813, 2011. 133, 134

[CM06] Judith A Chevalier and Dina Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, August 2006. 8, 120, 123

[Coh03] Jacob Cohen. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Inquiry and Pedagogy Across Diverse Contexts Series. Lawrence Erlbaum Associates, Abingdon, UK, 2003. 131

[Col03] Debbie Collins. Pretesting survey instruments: An overview of cognitive methods. *Quality of Life Research*, 12:229–238, 2003. 124, 125

[CS06] Anna L. Cox and Mariana M. Silva. The role of mouse movements in interactive search. In *Proceedings of the Twenty-Eighth Annual Meeting of the Cognitive Science Society (CogSci2006)*, pages 1156–1161, Austin, USA, 2006. Cognitive Science Society. 100

[Csa10] Gabor Csardi. *Network Analysis and Visualization*. `http://igraph.sourceforge.net/`, 0.5.4-1 edition, August 2010. Package 'igraph'. 48

[Del03] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, October 2003. 123

[DMKSD06] Benedetto De Martino, Dharshan Kumaran, Ben Seymour, and Raymond J. Dolan. Frames, biases, and rational decision-making in the human brain. *Science*, 313(5787):684–687, 2006. 126

[DR01] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 57–66, New York, USA, 2001. ACM. 49

[EBB10] Ilham Esslimani, Armelle Brun, and Anne Boyer. Detecting leaders in behavioral networks. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM '10)*, pages 281–285, Washington, DC, USA, 2010. IEEE Computer Society. 43

[EF70] John W. Essam and Michael E. Fisher. Some basic definitions in graph theory. *Reviews of Modern Physics*, 42(2):272–288, April 1970. 46

[Eul36] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 8:128–140, 1736. 46

[fac13] Facebook reports fourth quarter and full year 2012 results, January 2013. `http://investor.fb.com/releasedetail.cfm?ReleaseID=736911`, accessed: June 12, 2013. 43

[FASJ00] Melissa L. Finucane, Ali Alhakami, Paul Slovic, and Stephen M. Johnson. The affect heuristic in judgments of risks and benefits. *Journal of Behavioral Decision Making*, 13(1):1–17, 2000. 125

[FK78] Richard Herbert Franke and James D. Kaul. The hawthorne experiments: First statistical interpretation. *American Sociological Review*, 43(5):623–643, 1978. 100, 105, 128

[FKS97] Josef Fink, Alfred Kobsa, and Jörg Schreck. Personalized hypermedia information provision through adaptive and adaptable system features: User modelling, privacy and security issues. In *IS&N '97: Proceedings of the Fourth International Conference on Intelligence and Services in Networks*, pages 459–467, Berlin, Germany, 1997. Springer. 39

[Fre78] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978. 48

[FW85] William Friedman and Arnold Wilkins. Scale effects in memory for the timeof events. *Memory & Cognition*, 13:168–175, 1985. 125

[GA10] Qi Guo and Eugene Agichtein. Towards predicting web searcher gaze position from mouse movements. In *CHI EA '10: Proceedings of the 28th of the International Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3601–3606, New York, USA, 2010. ACM. 101

[GH06] Scott A. Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006. 27, 164

[GI07] Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: Predicting the usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce (ICEC '07)*, pages 303–310, New York, USA, 2007. ACM. 120

[GS90] Paul E. Green and V. Srinivasan. Conjoint analysis in marketing: New developments with implications for research and practice. *Journal of Marketing*, 54(4):3–19, 1990. 131

[Gun12] Vic Gundotra. Google+: Communities and photos, December 2012. `http://googleblog.blogspot.de/2012/12/google-communities-and-photos.html`, accessed: June 12, 2013. 43

[HK10] Natascha Hoebel and Stanislav Kreuzer. Cord: A hybrid approach for efficient clustering of ordinal data using fuzzy logic and self-organizing maps. In *6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*, Porto, Portugal, April 2010. INSTICC Press. 34

[HKTZ06a] Natascha Hoebel, Sascha Kaufmann, Karsten Tolle, and Roberto V. Zicari. The design of gugubarra 2.0: A tool for building and managing profiles of web users. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 317–320, Washington, DC, USA, December 2006. IEEE Computer Society. 17

[HKTZ06b] Natascha Hoebel, Sascha Kaufmann, Karsten Tolle, and Roberto V. Zicari. The gugubarra project: Building and evaluating user profiles for visitors of web sites. In *HotWeb 2006 - First IEEE Workshop on Hot Topics in Web Systems and Technologies*, pages 1–7, Washington, DC, USA, November 2006. IEEE Computer Society. VI, 4, 5, 17, 18, 20, 59, 71, 85

[HKW01] Rinus Haaijer, Wagner Kamakura, and Michel Wedel. The no-choice alternative in conjoint choice experiments. *International Journal of Market Research*, 43(1):93–106, 2001. 132

[HLCK05] F. Harper, Xin Li, Yan Chen, and Joseph Konstan. An economic model of user rating in an online recommender system. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling 2005*, volume 3538 of

Lecture Notes in Computer Science, pages 149–149. Springer, Berlin, Germany, 2005. 123

[HLS11] Nan Hu, Ling Liu, and Vallabh Sambamurthy. Fraud detection in online consumer reviews. *Decision Support Systems*, 50(3):614–626, January 2011. 120

[HMS+09] Natascha Hoebel, Naveed Mushtaq, Clemens Schefels, Karsten Tolle, and Roberto V. Zicari. Introducing zones to a web site: A test based evaluation on semantics, content, and business goals. In *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing (CEC2009)*, pages 265–272, Washington, DC, USA, July 2009. IEEE Computer Society. 45, 100, 217

[Hoe11] Natascha Hoebel. *User Interests and Behavior on the Web: Measurements and Framing Strategies*. PhD thesis, Goethe-University Frankfurt am Main, 2011. 22, 23, 24, 103, 131, 191

[HPZ06] Nan Hu, Paul A. Pavlou, and Jennifer Zhang. Can online reviews reveal a product's true quality? empirical findings and analytical modeling of online word-of-mouth communication. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC '06)*, pages 324–330, New York, USA, 2006. ACM. 123

[HSDA07] Jan Hartmann, Alistair Sutcliffe, and Antonella De Angeli. Investigating attractiveness in web user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 387–396, New York, USA, 2007. ACM. 126

[HZ08] Natascha Hoebel and Roberto V. Zicari. Creating user profiles of web visitors using zones, weights and actions. In *Tenth IEEE Conference On E-Commerce Technology (CEC 2008) And The Fifth Enterprise Computing, E-Commerce And E-Services (EEE 2008)*, pages 190–197, Washington, DC, USA, July 2008. IEEE Computer Society. 13, 14, 43, 99

[HZ11] Natascha Hoebel and Darius Zumstein. Trust and privacy in the web: Do internet users feel monitored on websites? In *ACM WebSci'11: Proceedings of the 3rd International Conference on Web Science*, pages 1–4, Southampton, England, 2011. The Web Science Trust. 16

[HZP09] Nan Hu, Jie Zhang, and Paul A. Pavlou. Overcoming the j-shaped distribution of product reviews. *Communications of the ACM*, 52(10):144–147, October 2009. 123

[IJK08] Raghuram Iyengar, Kamel Jedidi, and Rajeev Kohli. A conjoint approach to multipart pricing. *Journal of Marketing Research*, 45(2):195–210, 2008. 133

[Ill07] Patrick Illinger. Mit kameras und peilsendern erforschen völkerkundler eines der größten mysterien der wissenschaft: Menschen beim einkaufen. *Süddeutsche Zeitung Wissen*, 14:66–77, January 2007. 16

[Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, February 1912. 48

[Jaz07] Mehdi Jazayeri. Some trends in web application development. In *FOSE '07: 2007 Future of Software Engineering*, pages 199–213, Washington, DC, USA, 2007. IEEE Computer Society. 41, 164

[JF03] Radu Jurca and Boi Faltings. An incentive compatible reputation mechanism. In *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1026–1027, New York, USA, 2003. ACM. 30

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, New York, USA, 2005. ACM. 39

[JHW07] Seikyung Jung, Jonathan L. Herlocker, and Janet Webster. Click data as implicit relevance feedback in web search. *Information Processing and Management: an International Journal*, 43(3):791–807, 2007. 15, 102

[JK08] Jia Jia-Kern. Analyse von datenschutzrechtlichen fragen bezüglich benutzerprofilbildung im internet. bachelor thesis, Goethe-University Frankfurt am Main, August 2008. 16

[Jun94] Dieter Jungnickel. *Graphen, Netzwerke und Algorithmen*. BI-Wissenschaftsverlag, Mannheim, Germany, 3th edition, 1994. 46

[Kan84] David E. Kanouse. Explaining negativity biases in evaluation and choice behavior: Theory and research. *Advances in Consumer Research*, 11:703–708, 1984. 126

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, September 1999. 48, 52

[KLL04] Dae-Won Kim, Kwang H. Lee, and Doheon Lee. Fuzzy clustering of categorical data using fuzzy centroids. *Pattern Recognition Letters*, 25(11):1263–1271, 2004. 34

[LLYT05] Kwei-Jay Lin, Haiyin Lu, Tao Yu, and Chia-en Tai. A reputation and trust management broker framework for web applications. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 262–269, Washington, DC, USA, 2005. IEEE Computer Society. 39

[Mar11] Jeff Marcus. Rgraphviz. Presentation, 2011. `http://files.meetup.com/1781511/RgraphViz.ppt`, accessed: June 12, 2013. 50

[Mcf74] Daniel Mcfadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, USA, 1974. 131

[ME95] David Maltz and Kate Ehrlich. Pointing the way: Active collaborative filtering. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 202–209, New York, USA, 1995. ACM. 40, 58, 59, 71

[MHM06] Jamie Murphy, Charles Hofacker, and Richard Mizerski. Primacy and recency effects on clicking behavior. *Journal of Computer-Mediated Communication*, 11(2):522–535, 2006. 126

[MJ09] Helen Margetts and Peter John. Experiments for web science: Examining the effect of the internet on collective action. In *Proceedings of the WebSci'09: Society On-Line*, Southampton, England, 2009. The Web Science Trust. 126

[ML01] Florian Mueller and Andrea Lockerd. Cheese: Tracking mouse movement activity on websites, a tool for user modeling. In *CHI '01: CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 279–280, New York, USA, 2001. ACM. 103

[MSB00] Sandra J. Milberg, H. Jeff Smith, and Sandra J. Burke. Information privacy: Corporate management and national regulation. *Organization Science*, 11(1):35–57, January 2000. 16

[MWTZ04] Naveed Mushtaq, Peter Werner, Karsten Tolle, and Roberto V. Zicari. Building and evaluating non-obvious user profiles for visitors of web sites. In *IEEE Conference on E-Commerce Technology (CEC 2004)*, pages 9–15, Washington, DC, USA, July 2004. IEEE Computer Society. 13, 18, 19, 43, 99

[NA08] Danil Nemirovsky and Konstantin Avrachenkov. Weighted pagerank: Cluster-related weights. In *Proceedings of The Seventeenth Text REtrieval Conference (TREC 2008)*, Gaithersburg, USA, November 2008. National Institute of Standards and Technology (NIST). 48

[OH08] Kieron O'Hara and Wendy Hall. Web science. *Association of Learning Technologies Newsletter*, 12, May 2008. VI, VII, 4, 5, 191

[O'R05] Tim O'Reilly. What is web 2.0? design patterns and business models for the next generation of software, September 2005. `http://www.oreillynet.com/lpt/a/6228`, accessed: May 15, 2012. 119

[Pal12] Georgios Paliouras. Discovery of web user communities and their role in personalization. *User Modeling and User-Adapted Interaction*, 22(1-2):151–175, 2012. 44

[PB97] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997. 39

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, pages 1389–1401, November 1957. 52

[PSD02] Josep M. Pujol, Ramon Sangüesa, and Jordi Delgado. Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '02): Part 1*, pages 467–474, New York, USA, 2002. ACM. 48

[RBLK10] Christoph Riedl, Ivo Blohm, Jan Marco Leimeister, and Helmut Krcmar. Rating scales for collective intelligence in innovation communities: Why quick and easy decision making does not get it right. In *Proceedings of* *the International Conference on Information Systems (ICIS 2010)*, Atlanta, USA, 2010. Association for Information Systems. 124

[RFAS08] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI '08: CHI '08 Extended Abstracts on Human Factors in Computing Systems*, pages 2997–3002, New York, USA, 2008. ACM. 101

[RN10] Marko A. Rodriguez and Peter Neubauer. Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41, August 2010. 46

[Sch95] Uwe Schöning. *Logik für Informatiker*. Reihe Informatik. Spektrum Akademischer Verlag, Heidelberg, Germany, 4th edition, 1995. 35

[Sch06] Clemens Schefels. Serving xcerpt to the web. Diplomarbeit/diploma thesis, Institute of Computer Science, Ludwig-Maximilians-University Munich, 2006. 215, 217

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007. 47

[Sch12] Clemens Schefels. How to find important users in a web community? mining similarity graphs. In *Proceedings of the First International Conference on Data Analytics (DATA ANALYTICS 2012) / NexTech 2012*, pages 10–17. International Academy, Research and Industry Association (IARIA), 2012. 43, 57, 216, 217

[Sch13] Clemens Schefels. Computing user importance in web communities by mining similarity graphs. *International Journal On Advances in Internet Technology*, 6(1 & 2):79–89, 2013. 43, 217

[SES12] Clemens Schefels, Sven Eschenberg, and Christian Schöneberger. Behavioral analysis of registered web site visitors with help of mouse tracking. In *Proceedings of the 14th IEEE International Conference on Commerce and Enterprise Computing (CEC2012)*, pages 33–40, Los Alamitos, USA, 2012. IEEE Computer Society Press. 99, 217

[SL07] Shahana Sen and Dawn Lerman. Why are you telling me this? an examination into negative consumer reviews on the web. *Journal of Interactive Marketing*, 21(4):76–94, 2007. 123, 124, 126

[Slo92] John A. Sloboda. *Cognitive Bases of Musical Communication*, chapter Empirical Studies of Emotional Response to Music, pages 33–46. American Psychological Association, Washington, US, 1992. 125

[Sol10] Wahid Solaimankhel. Explizites user feedback für gugubarra 3.0. bachelor thesis, Goethe-University Frankfurt am Main, August 2010. 24

[SS11] E. Isaac Sparling and Shilad Sen. Rating: How difficult is it? In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 149–156, New York, USA, 2011. ACM. 127

[SSRC+10] Michal Shmueli-Scheuer, Haggai Roitman, David Carmel, Yosi Mass, and David Konopnicki. Extracting user profiles from large scale data. In *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud (MDAC '10)*, pages 4:1–4:6, New York, USA, 2010. ACM. 40

[SWHZ13] Clemens Schefels, Ines Weimer, Natascha Hoebel, and Roberto V. Zicari. How web based rating systems influence user's choice? 2013. In Submission. 119, 217

[SZ10] Clemens Schefels and Roberto V. Zicari. A framework analysis for managing explicit feedback of visitors of a web site. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010)*, pages 481–488, New York, USA, November 2010. ACM. 29, 100, 217

[SZ12] Clemens Schefels and Roberto V. Zicari. A framework analysis for managing feedback of visitors of a web site. *International Journal of Web Information Systems (IJWIS)*, 8(1):127–150, 2012. 29, 57, 100, 217

[TCC04] Roger Tourangeau, Mick P. Couper, and Frederick Conrad. Spacing, position, and order: Interpretive heuristics for visual features of survey questions. *Public Opinion Quarterly*, 68(3):368–393, 2004. 126

[TCC07] Roger Tourangeau, Mick P. Couper, and Frederick Conrad. Color, labels, and interpretive heuristics for response scales. *Public Opinion Quarterly*, 71(1):91–112, 2007. 135, 142, 148

[TH07] Luis A. Leiva Torres and Roberto Vivo Hernando. (smt) real time mouse tracking registration and visualization tool for usability evaluation on websites. In Pedro Isaias, Miguel Baptista Nunes, and Joao Barroso, editors, *Proceedings of the IADIS International Conference on WWW/Internet*, volume 1, pages 187–192, Lisbon, Portugal, 2007. IADIS Press. 101

[TK74] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974. 125

[TRR00] R. Tourangeau, L.J. Rips, and K. Rasinski. *The Psychology of Survey Response*. Cambridge University Press, Cambridge, UK, 2000. 124, 125, 127, 157

[VSL07] Paul Van Schaik and Jonathan Ling. Design parameters of rating scales for web sites. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1):419–432, 2007. 41

[VW07] Thomas Vander Wal. Folksonomy coinage and definition, February 2007. `http://www.vanderwal.net/folksonomy.html`, accessed: September 22, 2008. 27, 164

[WBW90] T.H. Wonnacott, C.D. Block, and R.J. Wonnacott. *Introductory Statistics for Business and Economics*. John Wiley & Sons, Ltd., Hoboken, New Jersey, USA, 4th edition, 1990. 133

[WH06] Birgit Weischedel and Eelko K. R. E. Huizingh. Website optimization with web metrics: a case study. In *Proceedings of the 8th International Conference on Electronic Commerce: The New E-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet (ICEC '06)*, pages 463–470, New York, USA, 2006. ACM. 15, 102

[Wil79] Robin J. Wilson. *Introduction to Graph Theory*. Longman, London, UK, 1979. 46

[WJR01] Ryen W. White, Joemon M Jose, and I. Ruthven. Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 534–538, Gaithersburg, USA, 2001. National Institute of Standards and Technology (NIST). 39

[WO81] Robert A. Westbrook and Richard L. Oliver. Developing better measures of consumer satisfaction: Some preliminary results. *Advances in Consumer Research*, 8(1):94–99, 1981. 120

[XB08] Lan Xia and Nada Nasr Bechwati. Word of mouse: The role of cognitive personalization in online consumer reviews. *Journal of Interactive Advertising*, 9(1):3–13, 2008. 123, 124

[YJSH08] Yeliz Yesilada, Caroline Jay, Robert Stevens, and Simon Harper. Validating the use and role of visual elements of web pages in navigation with an eye-tracking study. In *Proceeding of the 17th international conference on World Wide Web (WWW '08)*, pages 11–20, New York, USA, 2008. ACM. 40

[YL03] Lan Yi and Bing Liu. Web page cleaning for web mining through feature weighting. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 43–48, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. 45

[ZZ06] Philip Zigoris and Yi Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 397–404, New York, USA, 2006. ACM. 39

Curriculum Vitae

Dipl.-Inform. Clemens Schefels

clemens@schefels.de
Goethe-University Frankfurt am Main
FB12 - Institute of Computer Science
Databases and Information Systems (DBIS)
Robert-Mayer-Straße 10
D-60325 Frankfurt am Main
Germany

Clemens Schefels, Dipl.-Inform., born 1977 in Munich, studied from 2000 to 2006 Computer Science with Statistics as a special subject at the Ludwig-Maximilians-University Munich. Here, he wrote his Diploma thesis [Sch06] about the integration of Web Services into a declarative XML query and transformation language based on logic programming concepts.

In May 2007, he started as a research assistant at the Goethe-University of Frankfurt in the work group of Databases and Information Systems (DBIS) and started his PhD thesis in 2009.

His research interests are: Explicit and implicit user feedback, user profiling, user behavior, web technologies, database systems, information systems, eCommerce, web services, web analytics, data analytics, and cognitive heuristics.

PERSÖNLICHE ANGABEN

Vorname	Clemens
Nachname	Schefels
Geburtsdatum	21.10.1977
Geburtsort	München, Deutschland

AUSBILDUNG

ab 2009	Doktorand am Lehrstuhl für Datenbanken- und Informationssysteme der Goethe-Universität Frankfurt am Main
2000 - 2006	Diplomstudiengang Informatik mit Nebenfach Statistik an der Ludwig-Maximilians-Universität München
1998 - 1999	Wehrersatzdienst im Zitaheim München
1988 - 1998	Pestalozzi-Gymnasium München
1984 - 1988	Rotbuchen Grundschule München

BERUFLICHER WERDEGANG

ab 2007	Wissenschaftlicher Mitarbeiter am Lehrstuhl für Datenbanken- und Informationssysteme der Goethe-Universität Frankfurt am Main
2004 - 2006	Wissenschaftliche Hilfskraft am Lehrstuhl für Programmier- und Modelliersprachen der Ludwig-Maximilians-Universität München
2004 - 2006	Freier Mitarbeiter bei Lorenz-it (München)
2004 - 2005	Freier Mitarbeiter Referat IIIA3 der Ludwig-Maximilians-Universität München
2003 - 2004	Freier Mitarbeiter Referat Internet der Ludwig-Maximilians-Universität München

AUSZEICHNUNGEN

Best Paper Award	Für den Konferenz Beitrag: C. Schefels, „How to Find Important Users in a Web Community? Mining Similarity Graphs" [Sch12], präsentiert auf der First International Conference on Data Analytics (DATA ANALYTICS 2012)

VERÖFFENTLICHUNGEN

2013	Clemens Schefels, Ines Weimer, Natascha Hoebel, Roberto V. Zicari „How Web Based Rating Systems Influence User's Choice?" In Submission, [SWHZ13]
2013	Clemens Schefels „Computing User Importance in Web Communities by Mining Similarity Graphs" In: International Journal On Advances in Internet Technology, Volume 6 - Number 1 & 2, Int. Academy, Research and Industry Association, [Sch13]
2012	Clemens Schefels „How to Find Important Users in a Web Community? Mining Similarity Graphs" In: Proceedings of The First International Conference on Data Analytics (DATA ANALYTICS 2012 / NexTech 2012), Barcelona, Spain, 23th-28th of September 2012, [Sch12]
2012	Clemens Schefels, Sven Eschenberg, Christian Schöneberger „Behavioral Analysis of Registered Web Site Visitors with Help of Mouse Tracking" In: Proceedings of the 14th IEEE Conference on Commerce and Enterprise Computing (CEC 2012), Hangzhou, China, 9th-11th of September 2012, [SES12]
2012	Clemens Schefels, Roberto V. Zicari „A Framework Analysis for Managing Explicit Feedback of Visitors of a Web Site" In: International Journal of Web Information Systems (IJWIS), Volume 8 - Issue 1, Emerald, [SZ12]
2010	Clemens Schefels, Roberto V. Zicari „A Framework Analysis for Managing Explicit Feedback of Visitors of a Web Site" In: Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (iiWAS2010), Paris, France, 8th-10th of November 2010, [SZ10]
2009	Natascha Hoebel, Naveed Mushtaq, Clemens Schefels, Karsten Tolle, Roberto V. Zicari „Introducing Zones to a Web Site: A Test Based Evaluation on Semantics, Content, and Business Goals" In: Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC09), Vienna, Austria, 20th-23th of July 2009, [HMS^{+}09]
2007	Clemens Schefels „Serving Xcerpt to the Web" Diplomarbeit, Ludwig-Maximilians-Universität München, [Sch06]

REVIEWS UND KONFERENZ-ORGANISATION

Program Committee	The Second International Conference on Data Analytics (DATA ANALYTICS 2013)
Program Committee	The First International Conference on Building and Exploring Web Based Environments (WEB 2013)
Program Committee	The Second International Conference on Advanced Collaborative Networks, Systems and Applications (COLLA 2012)
Program Committee und Local Organization	3rd International Conference on Objects and Databases (ICOODB2010)
Review	17th European Conference on Information Systems (ECIS2009)

MITARBEIT IN DER AKADEMISCHEN SELBSTVERWALTUNG

2011	Unterstützung bei der Girls-LAN-Party (Frauenförderungsprogramm)
ab 2009	Mitglied im Lehr- und Studienausschuss der Informatik
2009	Unterstützung beim Girls-Day (Frauenförderungsprogramm)
ab 2007	Mitglied im Bauausschuss

MITARBEIT BEI LEHRVERANSTALTUNGEN

SoSe 2012	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
WiSe 2011/12	Vorlesung: „Datenbanksysteme I“
SoSe 2011	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
WiSe 2010/11	Vorlesung: „Datenbanksysteme II“
	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
WiSe 2009/10	Vorlesung: „Datenbanksysteme I“
	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
SoSe 2009	Vorlesung: „Principles of E-Commerce I: Business and Technology“
	Vorlesung: „Grundlagen der Programmierung 2“
	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
WiSe 2008/09	Vorlesung: „Principles of E-Commerce II: E-Services & Web Business Models“
	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
SoSe 2008	Vorlesung: „Datenbanksysteme II“
	Seminar: „Informationssysteme: Web-Technologien für E-Commerce“
	Vorlesung: „Grundlagen der Programmierung 2“
WiSe 2007/08	Vorlesung: „Datenbanksysteme I“
SoSe 2007	Vorlesung: „Grundlagen der Programmierung 2“