TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

DIPLOMARBEIT

Investigating Web Usage Data with Visual and Explorative Analysis Methods

Ausgeführt am Institut für

Softwaretechnik und interaktive Systeme
der Technischen Universität Wien

unter der Anleitung von *Ao.Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Silvia Miksch*

durch

Stefan Schnabl
Laxenburgerstr. 59/12
A-1100 Wien
Österreich

Datum

Stefan Schnabl

Contact Information:

student:	Stefan Schnabl
matriculation number:	0306110
address:	Laxenburgerstr. 59/12
	A-1100 Vienna, Austria
email:	stefan.schnabl@gmail.com

University Supervisor:
a.o. Univ. Prof. Mag. Dr. Silvia Miksch

Institute of Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11 / 188
A-1040 Vienna, Austria
http://www.ifs.tuwien.ac.at

Abstract

The work presents methods to support maintainers and developers of web applications in the field of software engineering and business management with visualization of web usage data. Usage data from an exemplary web application is analyzed by using mostly existing and also newly developed visualization and interaction methods.

The general concept of control circuits is combined with the basic information flow in and around web applications. The resulting model is presented as the Web Information Cycle (WIC). By that we present an exhaustive model for the information flow trough and around web applications. Within the WIC we present the positions of the data generating process for the analyzed usage data as well as the point where the results of the analysis provide a benefit for clients.

For an adequate way of the data representation we present a selected set of existing visual and interaction methods. We put special focus on visual methods with a statistical background and describe the mosaic plot with its capabilities for the work. Moreover, we present a set of existing interaction methods and discuss their usefulness for the work. For a single task we discuss an individual approach and present a visualization method named *Logsnakes*. The method is implemented in R code for further use.

A proposed data model along with the visualization and interaction methods is furthermore described by UML diagrams. Those are the basis for interviews with experts where the concept is evaluated with respect to its possible application in practice. The results from the interviews are presented and discussed critically.

Kurzfassung

Die vorliegende Arbeit präsentiert Methoden zur Unterstützung von Entwicklern von Internet Applikationen im Bereich der Softwareentwicklung. Der Fokus liegt hierbei vor allem auf Unterstützung der Umsetzung der Geschäftsprozesse der Applikation durch Visualisierung. Dazu werden Nutzungsdaten einer exemplarischen Web Applikation mit existierenden und neu entwickelten Visualisierungs- und Interaktionsmtehoden analysiert.

Das bekannte Konzept eines Regelkreises wird durch Kombination mit dem Informationsfluss in und um eine Web Applikation zum Web Information Cycle (WIC) zusammengeführt. Damit wird ein umfassendes Konzept zur Darstellung des Informationsflusses in und um Web Applikationen vorgestellt. Innerhalb des WIC werden die relevanten Punkte für den datengenerierenden Prozess der Nutzungsdaten und deren Rückfluss in Form von Analysen im Detail untersucht. Der Rückfluss von Information in Form von Analysen stellt einen Vorteil gegenüber bekannten Methoden dar und bringt eine Verbesserung für die Entwickler und in weiterer Folge für die Nutzer der Web Applikation.

Um die Nutzungsdaten in geeigneter Form darstellen zu können präsentieren wir eine Auswahl an existierenden Visualisierungs- und Interaktionsmethoden. Spezieller Fokus wird hierbei auf Methoden mit einem statistischen Hintergrund gelegt, wobei wir den Mosaicplot im Detail darstellen. Auf praktische Verwendbarkeit der Methoden wird ebenfalls eingegangen. Für eine spezielle Anwendung wird eine individuelle Visualisierungstechnik mit Namen *Logsnakes* in R Code erstellt.

Ein vorgestelltes Datenmodell zusammen mit den Visualisierungs- und Interaktionsmethoden wird durch UML Diagramme beschrieben. Diese Diagramme sind die Basis für Experteninterviews in welchen die Konzepte mit Fokus auf den parktischen Einsatz evaluiert werden. Die Resultate der Interviews werden präsentiert und diskutiert.

Acknowledgements

Writing this thesis was a lot of hard work, but also provided a lot of fun. As I went on with putting the contents in shape and describing the topics, I caught myself wanting to improve the subjects and describing things in more detail. This is a natural consequence when I like what I am working on. Finishing such a work involves a lot of persons which assist and help to make things possible.

First and foremost I want to thank my parents Hildegard and Werner, who provided a background in which I could freely decide to attend university. This was one of the cornerstones of my academic development, which I appreciate a lot and do not take for granted.

I also want to thank Univ.-Prof. Dr. Silvia Miksch for the help and possibility to write this thesis and her encouragement for the topic in general.

Furthermore I want to thank Dr. Christian Lutsch for his assistance in proof reading the thesis as well as for his overall aid. I see hardly any chance my studies would have gone the way they are without him.

For the fruitful discussions which entered into chapter ,,Evaluation", I want to thank Univ.-Prof. Dr. Marcus Hudec from the University of Vienna and Jeffrey R. Horner, BSc from the Vanderbilt University in Nashville, Tennessee as well as Dipl.Ing. Erich Gstrein and Florian Kleedorfer from the SAT research studios. They answer for evaluating the DICE concept and providing great ideas and great feedback for my work.

Moreover I thank Leonhard Seyfang, BSc and Dipl.Ing. Manuel J. A. Eugster for the procreative discussions.

Stefan Schnabl, Vienna in July 2007

Contents

Chapter 1

Introduction

> The more I want to get something done, the less I call it work.
> - Richard Bach -

1.1 General Issues for the Reader

1.1.1 Terms and Definitions

term	**description**
abilities (of the DICE tool)	The term *abilities of the tool* addresses all visualization methods and interaction techniques presented within this work.
active usage recording	The recording of user behaviour is an integrated part of the web application. The collected data is heavily optimized for the task of the subsequent analysis.
analyst	The person who uses the DICE tool is called *analyst*. He or she derives hypothesis about possible hints for the enhancement of the web application.
data analysis	One or more periods in which the analyst uses the DICE tool to derive hypothesis.
data exploring process	The process the analyst has to go through to gain insight into the data and derive hypothesis.
data generating process	The (general) process which generates the data-basis used for analysis. In our case this is a *web application*.
data pool	The combination of an effective collection of usage data and an adequate data model leads to a *high quality data pool*.

DICE tool (or just ,,tool")	The DICE tool is a piece of software which enables the analyst to explore and visualize the dataspace recorded from the usage of a web application. It should provide an useful set of methods and techniques to get an initial overview of the data and insights into possible data relations.
gap of views	The gap between the developers point of view and the users point of view regarding the processes of the web application (see section 2.1).
maintainers view	The application seen from the maintainers point of view. This view represents the aim of the developers of the application.
report	The work product of the analyst. It confirms the hypothesis with images and extracts from the DICE tool.
session	The usage of a web application. The session starts with the login procedure and ends with the logout. It involves the usage of the web application.
usage data	The data produced by *active usage recording* (see *data pool*).
user	The user of the data generating web application. The person using the DICE tool is called *analyst*.
user view	The view the user has on the application of interest is the relevant one. Thus recorded usage data should reflect the view as seamlessly as possible.
web application or *web-based application* or *application*	The web application is one possible basis for the recording of usage data. Within this work, a web application will be the data generating process. The data recorded will be used for the exploration with the DICE tool. We will define a web application in section 2.2.1.
web information cycle	The abstract path of information from its creation over the publication within the web application (see section 2.3).
webpage	A single page delivered via the HTTP protocol and displayed in a web browser.
website	A linked network of webpages.

Table 1.1: Terms and Definitions of the work.

1.1.2 Example

Since the thesis is oriented towards practical problems and tasks I think it is essential to provide an example. It is very hard to present an example and, parallel, go on with a consistent thesis. To prevent the aggravation of having to go back and forth all the time, I try to ,,weave" the example into the text and point out important relations. The relevant blocks are marked by *gray background*.

Example - Introduction

The example for the thesis will be a web application which is an online music portal. Users of the portal can perform actions such as *rate*, *view*, or *buy*. The objects affected by this actions are music tracks and music related goods (such as ringtones).

The overall goal is to optimize the application for both, users and developers.

The implementation of a web application with *active usage recording* is described in detail in [Gstrein2005]. The document also contains the following:

- the URL address of the implementation (http://www.ericsson-mediasuite.com/),
- data collection issues,
- the data model of the implementation, and
- proposed questions which should be answered within the analysis.

1.1.3 Additional Material on CD

The thesis has a compact disc coming along with it including the following contents:

- `ROOT`: Am appropriate `readme.txt` file.
- `ROOT`: The thesis as pdf and dvi file.
- `/code`: The R code for investigating the data.
- `/examples`: Exemplary files (e.g., data files).
- `/grafics`: The graphics used in the thesis.
- `/uml`: UML diagrams.

1.2 Motivation

1.2.1 Description of the Problem

,,*Web-based software applications, which enable user interaction through web browsers, have been extremely successful.*" ([Bultan *et al.*2005])

The persons who maintain web-based software applications have a complicated task. There are methods and techniques to improve the web application in terms of usability and customer orientation. [Kappel *et al.*2004] shows us three phases of relevant web-usability engineering:

1. **requirement analysis**,
2. **concept and implementation**, and
3. **application**.

Within the third phase, *application*, [Kappel *et al.*2004] identifies the following techniques which are relevant for a user-centered approach:

1. **logfile-analysis**,
2. **server statistics**, and
3. **user feedback analysis**.

The first two techniques are commonly offered by modern web server systems. The latter can be realized either *offline* or *online*. The focus in this work clearly lies on *the investigation of online methods* for user feedback analysis which make use of visualization and interaction techniques. With these techniques one can improve the effectiveness of the processes the web application offers the users [Tiedtke *et al.*2002]. Those methods will be named *DICE concept* within this work. The DICE concept was experimentally implemented as a prototype.

The input data for the online methods should fully reflect the the users activities. This means, that the abstraction of the recorded data should give clear hints about the performed processes, the users actions, and the success states of those processes. The optimization of usage data with respect to the users behaviour can give a clear *exposition of the users perception of the web application*. This perception of the users can be compared to the intentions of the web application developers. This comparison can result into a difference which we will call *the gap of views* .

The basic model which we use to illustrate the gap of views is a control circuit. The control circuit shows, where the relevant information could *flow back* into the evaluation circle of the web application. In figure 1.1 the particular spots of interest in the circuit are marked with numbers one and two. These are the points where human interaction with the system takes place. The presentation of the concept as closed information flow has several competitive advantages (compare [Wikipedia2006c]):

- No a-priori loss of information.
- Awareness of the data generating processes and the *usage data* they generate.
- External data input is possible. Thus the model can be seen as a generalized approach to other models (e.g., [Spiliopoulou2000]).

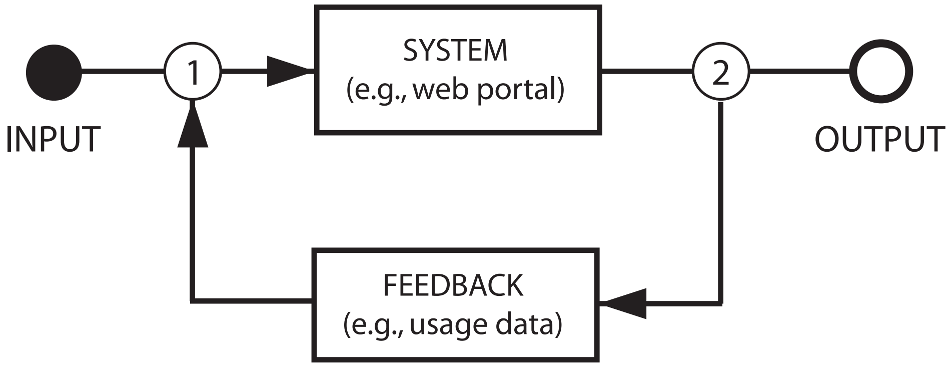


Figure 1.1: A control circuit following the DIN rules.

In point one the developer's / maintainer's view and the view of the user meet. The user's view is reflected by the collected usage data. With appropriate visualization methods *the* *gap between the views* should be emphasized as clear as possible. The reports generated by the DICE tool should deliver clear hints, where the web application could be optimized for a mutual benefit. These hints can be used as a basis for the enhancement of the web application such that the gap of views is reduced [Spiliopoulou2000].

On the other hand, spot two shows the point where the collection of the usage data takes place. The feedback data is generated in spot two. Thus, spot two is not really ,,a spot", but more an *integrated feature* throughout the whole application. The data collection can be done with respect to a certain data structure (see section 5.1.1) and with respect to the processes one wants to analyze. This integrated functions are called *active usage recording* and lead to high quality feedback data (see section 2.3.5).

The task of presenting the data in a clear and structured way is the basis for an exploration. To enable developers and maintainers, who know the underlying process of the application, to derive clear statements, is ,,not a trivial undertaking" ([Haigh and Megarity1998]). The DICE tool should exactly fulfill this need. Once the data is loaded, one can look on partly aggregated data in many ways providing the aid for a sharp and deep insight in the data. For some of the techniques used, a general benefit was shown by [Crad *et al.*1999] and [Shneiderman1996].

1.2.2 The Goal

The goal of DICE is *to enable people, who have little or no insight in the structure of the underlying data but in the data generating process, to derive hypothesis about interdependences inside the data.* This means that maintainers and developers can receive a structured overview of the recorded usage data. Additionally they can perform data analysis with integrated visual methods and interaction techniques. This can help to explore the data in a beneficial way.

The reports generated with the DICE tool cover and extend state-of-the-art web server statistics. These statistics generally reflect a technical point of view, whereas DICE wants to reflect the users view on the web applicaiton. To be able to visualise the users view on the web application, DICE combines visual methods with interaction techniques. The implementation furthermore takes respect to the information seeking mantra introduced by [Shneiderman1996] and supports *an iterative way of exploring the dataspace.*

The implementation of the DICE concept would be a piece of software, which can handle the proposed techniques and methods in a convenient way. This software has the name *DICE tool.* If data from a certain process is successfully loaded into the DICE tool, persons with knowledge about the data generating process can explore the effectiveness and/or the success of the intended process (e.g., a certain service of the web portal).

In detail, goals of the DICE tool can be verbalized as:

- The data exploration should be doable in an **iterative manner**, as suggested by the information seeking mantra[Shneiderman1996].

- The tool itself, especially its interface, should be easy to use. After a reasonable amount of learning time, **any person should be able to handle the tool efficiently**.

- It must be possible to apply more than one technique at a time. This generates a **variety of views on the problem**.

- The DICE tool should support **multiple views**, so that techniques such as linking and brushing can be used. These techniques help to get insight into dependencies between certain aspects of the data.

- The DICE tool should **assist developers and maintainers of web applications** to derive hints and hypothesis for further enhancement of these web applications.

1.3 Overview of the Thesis

This thesis is divided into the following main parts.

Chapter one and two of the work deal with the environment in which the tool is going to work. In addition chapter three shows the current development state of web usage analysis. It tries to give a short overview of the core areas where a tool like DICE could be used.

Chapter four will give an insight in the proposed methods we want to equip DICE with. We want to point out how much potential these methods have in theoretical means and within their implementation in the DICE tool. We describe the statistical methods and their theoretical background as well as state of the art methods in visualization. In addition we want to give an outlook on the potentially useful enhancements of the DICE tool.

Chapter five deals with the implementation of the DICE tool. Because we present a concept, this chapter will cover

- a project outline including a data model and the relevant actors,
- the concept of multiple views and its theoretical background,
- detailed use cases for the DICE tool, and
- possible architectures for the implementation.

In chapter six one can find the evaluation of the DICE concept. The evaluation is done by expert opinions on the topics of

- the DICE concept in general,
- the need for a tool like DICE in practice,
- possible architectures for an implementation, and
- potential shortcomings and challenges for the DICE concept.

The final chapter seven will extend the work for its results regarding the problem presented in chapter one as well as a brief summary of the work. It also shows ideas and possible enhancements of the concept for future work.

1.4 Conclusion

The DICE tool and the thesis are a combination. Its primary components are a practical and a theoretical part which work together mutually beneficial. The thesis tries to present a theoretical fundament for practical techniques. The result is an effective and high quality concept for a web usage analysis tool. An evaluation of the concept tries to ensure the value of the product.

The next chapter of the work will analyze the problem in detail and extend section 1.2.1 as well as section 1.2.

Chapter 2

Problem Analysis

> ” Put more succinctly, a Web application is software on the web! “
> - Bob Baxley -

2.1 Motivation

In addition to section 1.2, we want to give further argumentation why

- the analysis of data from web-based applications is important and
- why the analysis needs *visual* and *statistical data analysis methods.*

„As more organizations view the Web as an integral part of their operations and external communications, interest in the measurement and evaluation of Web site usage is increasing.” ([Haigh and Megarity1998])

The statement from [Haigh and Megarity1998] gives us a clear hint why measuring the success of web-based applications attracts an increasing audience. We find similar statements in [Bolz2001]. There are several reasons why the analysis of web applications can provide a better service to customers and users of the applications as well as their services. As [Spiliopoulou2000] puts it: „ ... conclusions are only valid if the users perceive the site and understand its services *as the designers have conceived them.*” This means that processes must not necessarily be understood by the users in the same way as by the developers and designers. We can identify a *gap of views* which reflects the quality and usability of a web application with respect to the different understanding of the processes. Of course, this gap should be minimized and, ideally, it should not be there at all.

According to these observations we can formalize our tasks by

- minimising the gap of and
- maximising the understanding of the processes within the application.

2.1.1 Preprocessing

To achieve the above mentioned goals, we must monitor user's activity paths along the processes. For this we should use an effective methodology (introduced as *active usage recording*) and store the data in an adequate data model (cf. section 5.1.1). This provides a high-quality data pool containing usage data which reflects the processes of the application.

2.1.2 Pattern Discovery and Analysis

To identify the patterns of interest in the data, we have to focus on relevant tools. We have an effective data pool which is optimized with respect to the processes inside the application. The next step is to identify patterns within the data. Visual and explorative data analysis methods have shown to be suitable tools for this kind of task. As a consequence we will focus on these methods to help us discovering patterns within the data. In addition, we use statistical methods in combination with interaction techniques. This provides a background which can be used for further statistical computation. For instance, data visualized with a mosaicplot could be tested for independence using a Chi-Square test statistic.

This would provide us with two elements on which we can base our hypothesis: a visual ,,image" and a theoretical test statistic. This can be seen as an advantage over pure visual methods. On the other hand, we can use the advantages offered by interaction methods (see section 4.1) to preprocess the data.

2.2 Web Application Analysis

To be able to identify the best and most appropriate techniques and methods for data analysis, we should first take a closer look at the data generating process. In our case the usage data will be generated by a *web-based application*. Although we are not able to fully generalize this processes and applications, we will try to point out aspects that are most common and can be used for further description.

A definition for a web application is given in [Baxley2002] and [Kappel *et al.*2004].

2.2.1 Definition of a Web Application

To find some common ground, we want to present a description of three types of commonly known websites:

1. **Static Websites:** The main point is that the content was created to be static. Updating this website always involves a lot of effort and time. Examples are:

   - Sites of high esthetic demand (e.g., pages presenting photography or arts) and
   - short information pages.

2. **(Semi -) Dynamic Websites:** Semi-dynamic websites are defined in [Liu *et al.*2000]. The basic model is a website which derives its contents from a database, which is updated regularly.

   On the contrary, [Liu *et al.*2000] speaks of dynamic websites where the site itself is created at runtime on basis of the actions performed. The website is presenting content which is delivered by a *content management system*.

Moreover, two definitions for dynamic web applications can be found on [Wikipedia2006a]. To summarize these statements I want to propose a definition for *dynamic websites*:

Definition - Dynamic Websites
„ *Dynamic websites are websites that are created or changed in a substantial manner on basis of external data at runtime, either derived with or without reloading of the page.* "

Examples are:

- Product presentation sites and
- picture galleries.

3. **Web Applications:** The user is known to the application he or she is using, there is a *logon procedure*. Information about the user is stored within the application and there are interaction possibilities within the application [Baxley2002]. Examples are[1]:

   - online booking systems,
   - marketplaces on the web,
   - financial services, or
   - information portals

During my research I found *web services* as a fourth type, described by [Bultan *et al.*2005] and on [Wikipedia2006d]. Web services deal with the communication *between applications* over the web and have a clear focus on the communication between software applications. As a consequence they are of little interest in this work. The focus of this work lies on the communication between the application and the user.

Example - Definition of the Exemplary Web Application

The web application, which serves as an example within this work, is described in detail in [Gstrein2005]. It was developed by the research studio Smart Agent Technologies.

The web application has the following properties which relate to the definition of a web application (following [Baxley2002]):

- **User logon:** The user has to log on to the system to perform relevant actions.
- **Process-oriented usage:** The usage is more than static preview of contents, its actions are modeled as processes within the application.
- **Information Storage:** The usage data is stored within the application and helps the user to perform tasks. Moreover, the application uses *active usage recording* for direct recording of the user behaviour with respect to the processes of the application.

The web application and its usage data can be fit into the proposed data model (see section 5.1.1). The actions which are possible have to be analyzed in detail to get the appropriate semantics for future analysis of the usage data.

[1] [Baxley2002] made a great overview on the basic understanding of web applications with lots of examples.

2.2.2 Actors

Another important part within the development of web applications, and of course their analysis, are the *actors* (or *roles*) which take action during the usage and maintenance of a web application.

1. **(Application) User:** The person who uses the web application to consume a service. This person creates an approach to the application. This *user view* is reflected by the usage data of the users sessions.

   Issues for the user are

   - the discrepance between the user expectation and the acting of the application (was introduced as *gap*),
   - trust in the quality and authenticity of the application, and
   - convenience when the user (re-)uses the application.

2. **Operating Company:** The operating company has most likely business issues to bring into the creation and maintenance of the web application. The data, in which the operating company is interested, might be recorded in other ways already [Keim2001], but might not be optimized for usage patterns.

   Issues for the operating company are

   - the return on investment of the application,
   - the consumer satisfaction, and
   - the potential growth of the application.

3. **Application Developer:** The application developers are having a special view on the application. They know exactly how things work within the application. As a result, it might be hard for them to relate to the problems a new user is facing.

   Issues for the application developer are

   - the stability of the system, and
   - the consistency of technical standards and implementations.

4. **Interface Developer:** Interface designer face a special task within the development of web applications. They have to bring the abstract process models into a form, so that users will know what to expect. This is one key aspect of successful application development in general [Baxley2002].

   Issues for the interface developer are

   - user control over the application, and
   - satisfaction when working with the application.

5. **Analyst:** The analyst will be the person who uses the DICE tool to derive hypothesis about the usage of the web application. The hypothesis will be used for the evolution of the website. Therefore, it is important that the analyst combines the views of the users and the maintainers as well as the view of the operating company. With this combination it is possible that he delivers products which help to improve the application.

   Issues for the interface analyst are

- high quality data,
- effective methodology for data integration,
- good possibility to combine interaction methods, or
- adequate visualization and statistical methods.

2.3 Web Information Cycle (WIC)

There is a classic ,,path of information” within web-based applications (or websites), which can be used for a general approach to the topic. We will call it the *web information cycle.* The following will argue the four basic steps of the web information cycle.

In recent years there has been a major impact on the way information is published on the web. According to [Notess2004] one reason was the development of tools and techniques for content management. The term for these tools and techniques is *content management systems, CMS*. One big change of professional CMS usage was a significant reduction of the publication time of web content.

Another important part of the WIC is user feedback. This feedback is considered to be essential for the evolution and the improvement of the application [Ankerst2000].

2.3.1 Content Creation - Step 1

The content for a web application is often defined by the operating company. This means that more aspects than just technical ones play an important role. Examples for these aspects are

- **business issues,** like the return on investment of the application or the number of products sold,
- **management issues** such as the modelling of the processes within the application follow certain business processes defined by the management, and
- **consumer issues** for better consumer satisfaction and, in relation, a more successful application .

We divide the above mentioned points in the following two groups

- **external data**, and
- **internal data**.

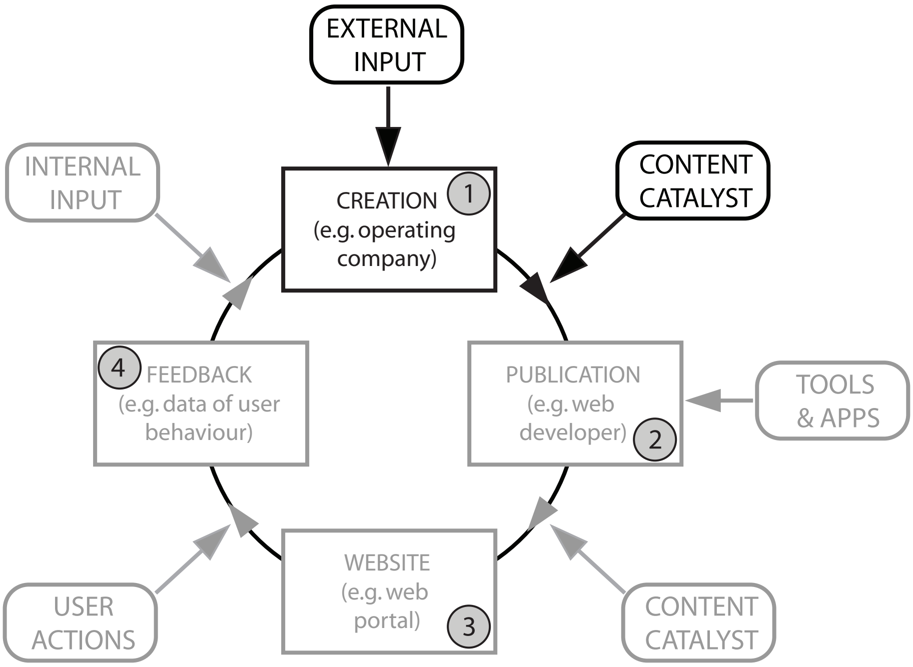

EXTERNAL INPUT
INTERNAL INPUT
CREATION (e.g. operating company) 1
CONTENT CATALYST
4 FEEDBACK (e.g. data of user behaviour)
PUBLICATION (e.g. web developer) 2
TOOLS & APPS
USER ACTIONS
WEBSITE (e.g. web portal) 3
CONTENT CATALYST

Figure 2.1: Step one of the web information cycle is the phase of content creation.

External Data

External data is the data the operating company does not retrieve from the existing application itself, but from a number of external items listed below (these items might also be other web applications). This data is reflected by *the content of the application* as shown in figure 2.1. Examples for external data are

- **advertisement** for other products or companies as well as for in-house products on the website,
- **product details** about products promoted on the website but not originally developed by the owners of this website,
- **legal regulations** controlling parts of processes in accordance with relevant legislation,
- **social aspects** which might influence the design and concept of the website to a significant extent, and
- **demographical aspects** reflecting the cultural background need to be taken into account by the application.

Example - External Data

To apply our example, typical *external content* for the web application could be

- **music album advertisement** for songs that are viewable and rateable within the web application,
- **details about songs and artists** such as a discography or personal information about the artist and his or her work, or
- **lists of products and prices**.

If the web application is *not* using *active usage recording*, then external data must cover more than the items mentioned above. If the operating company wants to know what users think about the application, or what they can do better, they have to gather data by external efforts. Constant monitoring of the sites' quality is very expensive and is considered to be too much effort [Spiliopoulou2000]. Possible necessary efforts are:

- **External Specialists:** The opinion of external specialists can, certainly, help to improve the tool. Nevertheless, it is a one-way activity and very cost intensive.
- **Customer Census:** This measurement is a very hard way of gathering data, because it needs the users to answer questions when they are not using the application. One has to find the appropriate people for the job and set up a propper experimental environment [Spiliopoulou2000].
- **Misusing other Sources:** In most cases the operating companies record some sort of data, because it is considered to inherit information valuable for business issues. Since this data mostly covers technical views on the application, it is hard to gain information about other views therefrom [Keim2001]. It could happen that an operating company does not know in detail what the differences between *active usage recording* and standard web logfiles are. If this is the case, they might try to get the needed information out of the logfiles.
- **Doing Nothing:** The worst case, from our point of view. The company does not evaluate the application by any kind.

Internal Data

Internal data is the type of data that is really interesting when describing the *web information cycle*, because it makes the cycle complete. When the application runs, it generates numerous data values, willingly or unwillingly. This data is the basis for continuous quality monitoring of the web application [Spiliopoulou2000]. It can be shown that most data pools are reflecting a technical point of view (e.g., the server side interest on the usage of the web application). Since this point of view gives little or no insight into the users activity it is not very useful as internal data in terms of the *web information cycle*.

Due to this fact there has to be *active usage recording*. An appropriate feedback function for the business processes has to be modelled. Not only does it have to pay respect to technical views on the problem, but also to the users point of view. This leads to a data pool which reflects the usage of the processes and services of the application.

Internal data is reflected by *the structure and the metaphors of the application*.

Example - Internal Data

To apply our example, typical *internal content* for the web application could be

- **User Actions:** Which actions did the user perform at what time?
  - **VIEW** Users look at tracks and assets.
  - **RATE** User rate music tracks.
  - **BUY** User buy assets related to music.
- **Process Barriers:** Spots where users drop out of processes before successful completion.
  - **Order Procedure:** For products (assets) related to music.
  - **Registration:** Dropouts during the registration process.

2.3.2 Content Publication - Step 2

The data which arrives at step two for *content publication*, has to be published within the web application as shown in figure 2.2. This is the spot where there has been a lot of development in the recent years [Bold and Neumeier2006]. The actual publication is done with respect to the complexity of the content.

Tools which are used when publishing online content are

- content management systems [Bold and Neumeier2006] (e.g., TYPO3),
- commercial web authoring tools,
- editors for web languages (e.g., Macromedia Dreamweaver),
- gallery applications (e.g., Gallery), and
- upload tools (e.g., FTP clients).

2.3.3 Content Catalyst

When internal and external data is brought together and put into the web application, it is certainly changed. This aspect will be described as *content catalysation* of the data stream. The extent, to which the data is influenced, is heavily determined by the tools that are used to transform the data for usage within the application. A flowchart for better understanding is presented in figure 2.3 and 2.4.

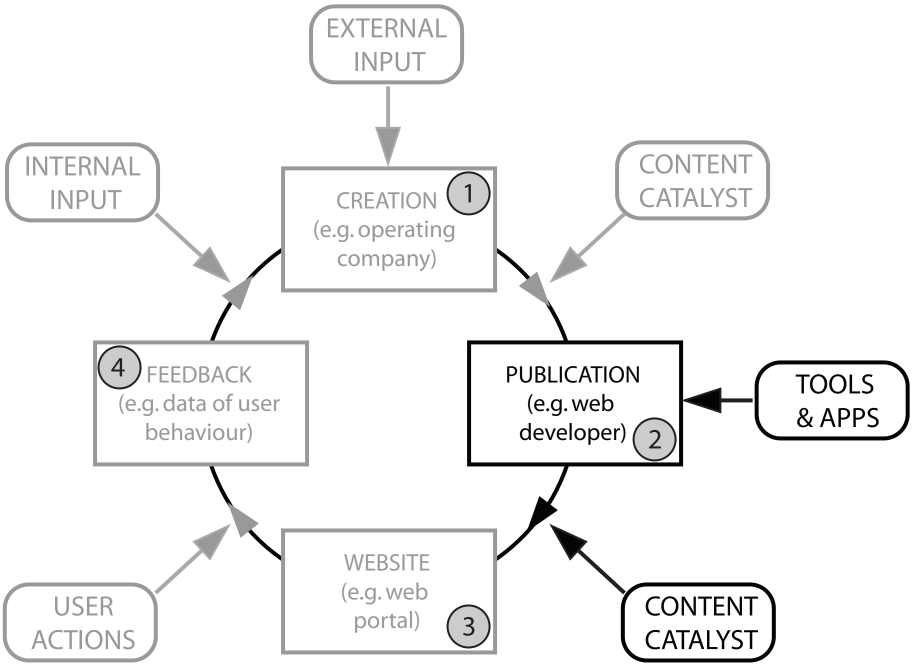

EXTERNAL INPUT
INTERNAL INPUT
CREATION (e.g. operating company)
1
CONTENT CATALYST
4
FEEDBACK (e.g. data of user behaviour)
PUBLICATION (e.g. web developer)
2
TOOLS & APPS
WEBSITE (e.g. web portal)
3
USER ACTIONS
CONTENT CATALYST

Figure 2.2: Step two of the web information cycle - content publication.

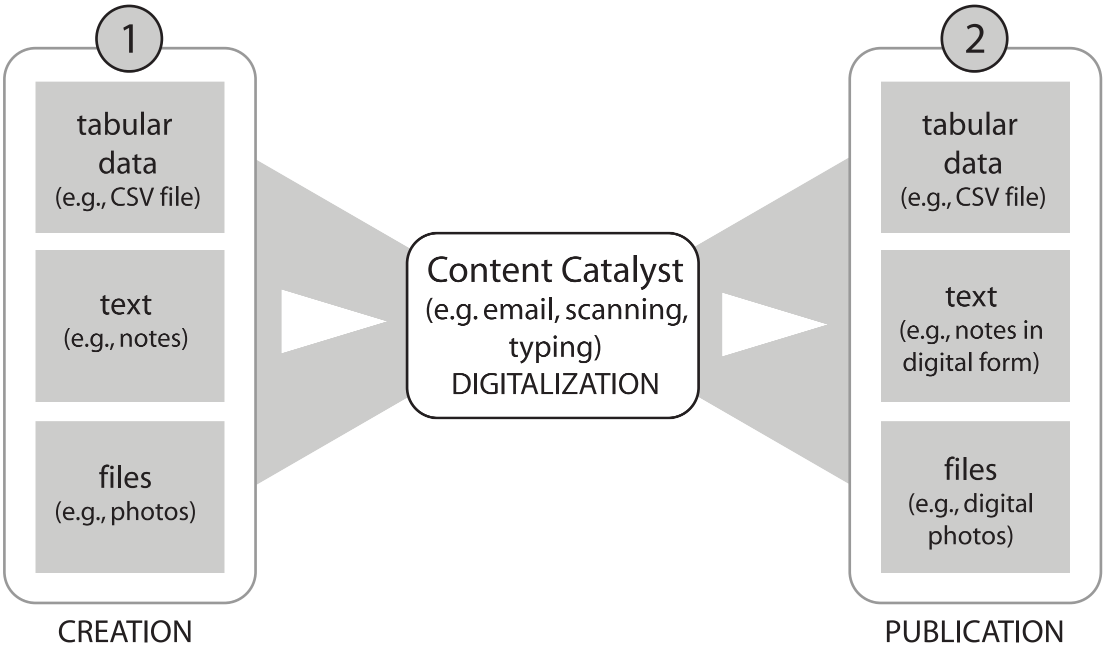

1
tabular data (e.g., CSV file)
text (e.g., notes)
files (e.g., photos)
CREATION
Content Catalyst
(e.g. email, scanning, typing)
DIGITALIZATION
2
tabular data (e.g., CSV file)
text (e.g., notes in digital form)
files (e.g., digital photos)
PUBLICATION

Figure 2.3: The first content catalyst within the web application cycle.

Example - Content Catalyst

Let us look now at some possible content catalyst for the *Web Information Cycle* between points 1 and 2 and between points 2 and 3. The following data has to be published within the web application:

1. **Product information data** as CSV file in tabular form, etc.
2. A census has resulted that the registration form contains the field *AGE*, which is disturbing many potential customers. The operating company agreed to **change the registration form** and skip the field *AGE*.
3. There are new **photos of products**, which ought to be put on the website as well.

The **catalyzing part** of this step (1 to 2 within the WIC) is the way the next actor (web developer) receives the information. Because whenever information is passed on, there is a possibility that it changes its form. For instance a text is copied into an email from MS Word and sent, then it has changed its form substantially. This also happens here.

1. **Product information data** is sent via email as attachment and has little or no change. The data was encapsulated within an enterprise resource planning system and exported to a CSV file. Here we have little or no change of the information.
2. The instruction for the **changing of the registration form** is sent via email, with little or no details about how to react on unforseen issues as
   - a shift in the constellation of the forms fields and a necessity for a complete redesign or
   - the technical consequences ot the changes.
3. The **photos of new products** were made with an analog camera and then scanned by the secretary. There a major catalyzation process happened. The web developer receives the photos as email attachment. Possible transformations may include
   - crop, scale, rotate or
   - change the light and contrast values

   before the images can be used as content in the web application.

The usage of tools to bring content into the web is most likely to change the form of the information. Actions like sizing, cropping, or editing and formatting can be summarized as *content authoring*. This means, the action applied change the technical state the information is in. Figure 2.4 schematically shows how the information is transformed.

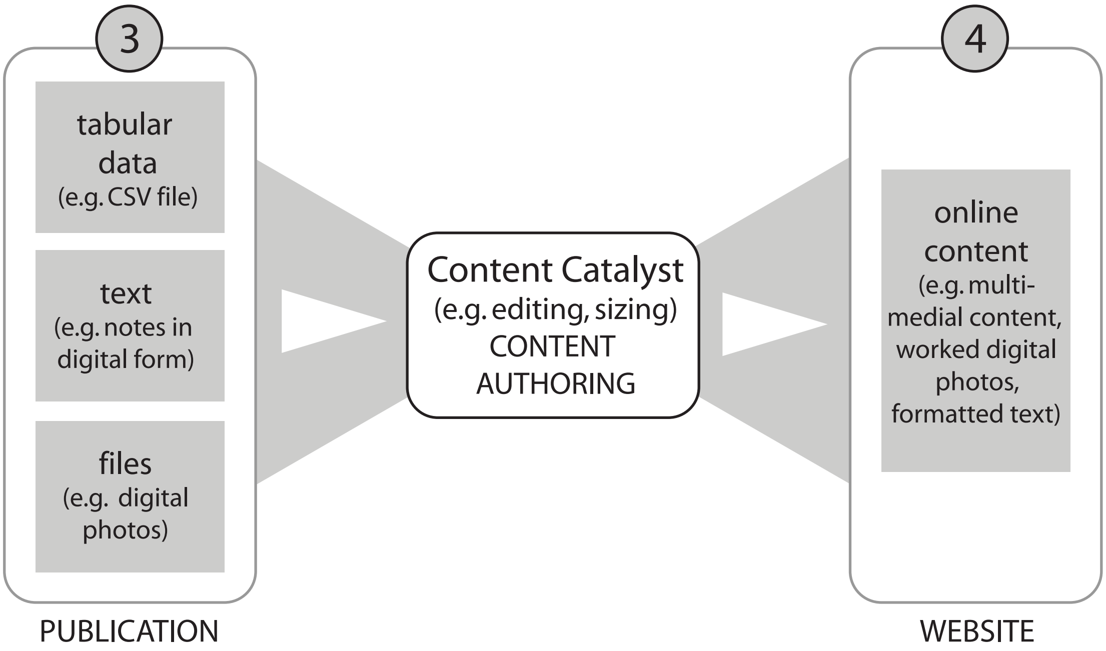

3
tabular data (e.g. CSV file)
text (e.g. notes in digital form)
files (e.g. digital photos)
PUBLICATION
Content Catalyst
(e.g. editing, sizing)
CONTENT AUTHORING
4
online content
(e.g. multi-medial content, worked digital photos, formatted text)
WEBSITE

Figure 2.4: The second content catalyst of the web information cycle.

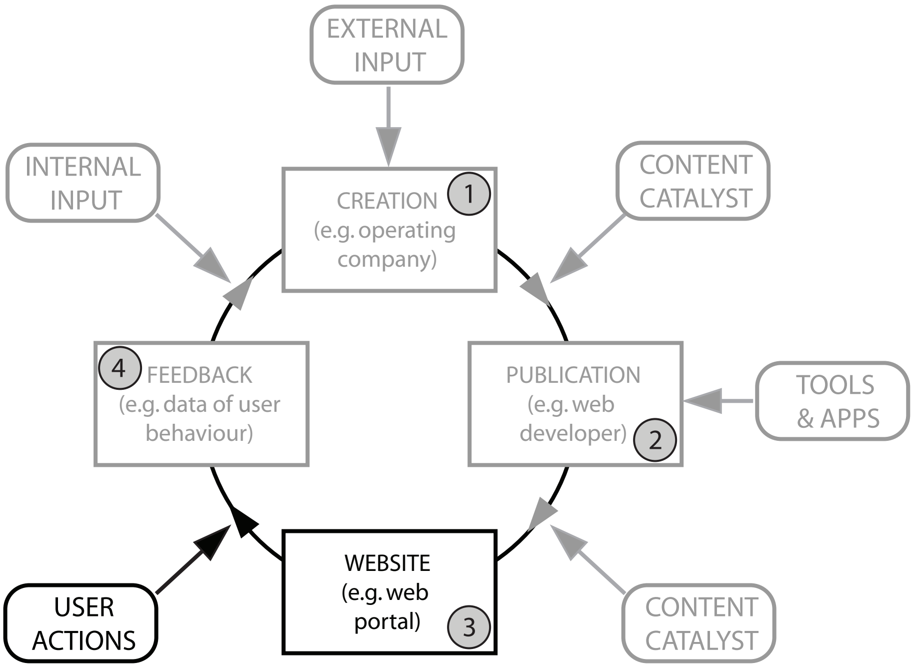

EXTERNAL INPUT
INTERNAL INPUT
CREATION (e.g. operating company) 1
CONTENT CATALYST
4 FEEDBACK (e.g. data of user behaviour)
PUBLICATION (e.g. web developer) 2
TOOLS & APPS
USER ACTIONS
WEBSITE (e.g. web portal) 3
CONTENT CATALYST

Figure 2.5: The third step of the web information cycle pictures the web application.

2.3.4 Web Application - Step 3

The content is now published and part of the application. At this place the user interacts with the software [Baxley2002] (see figure 2.5). Due to *active usage recording*, the application's processes produce data which maps the users behaviour.

Example - Web Application

The user can perform actions like:

- Register at the application.
- Perform actions within the application such as
  - VIEW tracks, artists and goods,
  - RATE tracks, artists and goods, and
  - BUY goods
- Manage his or her profile (e.g., change the registration data).
- Manage his or her previous sessions (e.g see results of previous sessions).

All this actions lead to a data pool. In the WIC we will call this data pool *user feedback*.

2.3.5 User Feedback - Step 4

The collected data which is relevant to the web information cycle, is the user feedback. This data is generated when the user interacts with the web application. The extent to which the data is recorded, in terms of quantity and quality (or complexity, e.g., the number of variables), is defined within the application as *active usage recording*. This procedure was mentioned by [Srivastava *et al.*2000] as *client side data collection*. [Srivastava *et al.*2000] points out two main advantages (compared to standard server-side data recording):

1. Problems with the **caching** of the client's browser: When cached content is displayed, there is no server request. There is no evidence for a request when in fact there was one. This can be omitted by the active recording of the users actions.
2. The **session identification**: It is hard to identify sessions within most common server log files. From our point of view it is necessary to have a clear login procedure. This makes sure the user (or user profile) is clearly assignable to a session.

The complexity of the recorded data can be influenced to any degree. It is controlled by the scripts which record the data. This means, that data recording could also be omitted. This case is also covered by the model of the web information cycle. Of course, recording no data at all would be very poor for the information cycle. As one can easily see, the cycle would lack a joint. The lack of content would have to be compensated entirely by external data.

The AWUSA implementation, which is mentioned in section 3, uses mainly common server log files as a basis for data mining and knowledge crystallisation [Tiedtke *et al.*2002]. The concept of this work goes a step further and thinks that the *active recording of data*, as done in the example application described in [Gstrein2005], is appropriate for modern web applications and their complex tasks.

„Ideally we would like to evaluate a site based on the data automatically recorded on it.” ([Spiliopoulou2000])

In [Tiedtke *et al.*2002] data preparation methods are proposed. These methods are needed because the data is not optimized with respect to the application's processes. Tasks like the following can be omitted due to *active usage recording* and due to the definition of a web application (see section 2.2.1):

- Elimination of unreal users (prevented by login procedure),
- identification of erroneous requests (can be treated within the active usage recording script), and
- data reduction (the data recorded is highly optimized per definition).

Additional data preparation methods can be useful for the DICE tool as well, since the active usage recording does not compensate them.

- Usability problems of the application,
- dead links, dead ends, or a bad structure of the application, or
- runtime errors (e.g., due to a server overload).

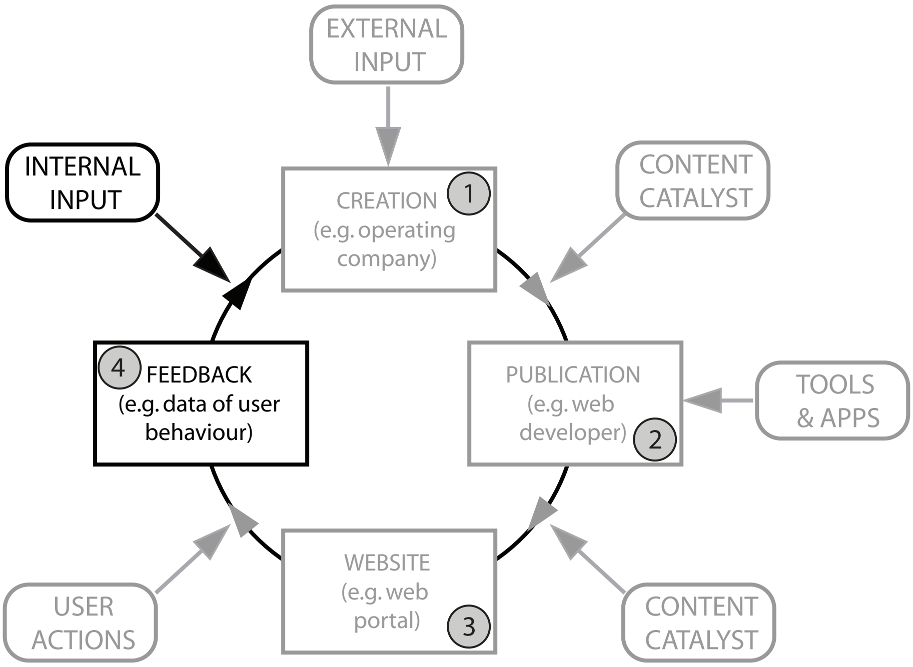

EXTERNAL INPUT
INTERNAL INPUT
CREATION (e.g. operating company)
1
CONTENT CATALYST
4
FEEDBACK (e.g. data of user behaviour)
PUBLICATION (e.g. web developer)
2
TOOLS & APPS
USER ACTIONS
WEBSITE (e.g. web portal)
3
CONTENT CATALYST

Figure 2.6: The fourth step of the web information cycle covers the user feedback.

ID	user ID	action ID	timestamp	success
1	2000100	10125 VIEW TRACK	2005-09-14 13:2	1
2	2000100	12152 VIEW ASSET	2005-09-14 16:4	1
3	2000100	12152 VIEW ASSET	2005-09-14 17:1	1
4	2000100	1617 VIEW TRACK	2005-09-14 17:1	1
5	2000100	7430 VIEW TRACK	2005-09-14 17:1	1
6	2000100	9449 VIEW ASSET	2005-09-14 17:1	1
7	2000100	9449 BUY ASSET	2005-09-14 17:2	1
8	2000100	12750 VIEW ASSET	2005-09-14 18:3	1
9	2000100	12750 BUY ASSET	2005-09-14 18:3	1
10	2000153	12750 VIEW ASSET	2005-09-15 10:1	1
	...			

Table 2.7: Simulated user feedback data from the exemplary web application.

Example - User Feedback Data

The data which we receive from *active usage recording* wants to generally answer the following: *Who has done what when with what outcome?*
The following hypothesis were generated by the analysis with the prototype of the DICE tool:

- There is no significant action after 1am.
- Profiled users are significantly more active than unprofiled users.
- The year of birth is not correlated to the action time.

A sample data frame, derived form the online music portal is presented in table 2.7. Simulated data, which is used for the example mosaicplots (e.g., figure 4.8), is listed in appendix C.

2.4 Conclusion

This chapter has recapitulated and extended the motivation for our work. Moreover, it has defined the term *web application* by way of examples. It provides a common ground for the field of study. By introducing the *web information cycle (WIC)*, it provides a framework for the data generating process. It acts as a generalization for existing work and extends it. As a substantial part of the WIC, *active usage recording* can provide a high quality data pool to secure a solid basis for data analysis.

There are numerous projects and products existing for web usage mining and there is an increasing demand for high quality analysis tools [Bolz2001]. The next chapter will present us with an overview of those projects which are related to the implementation of the DICE tool.

Chapter 3

State of the Art Web Usage Analysis Methods

> ... new communications protocols and servers may shed more light on users and usage.
>
> - Susan Haigh and Janette Megarity -

3.1 General

In this section we will try to bring up methods within the fields of *web mining* and *web usage analysis*. We want to point out the aspects in which the approach of *active usage recording* (and the DICE tool) tries to enhance the quality and ability of the performing analysis methods. The aspects cover the process of *data recording* as well as data *visualization* and *interaction*.

3.2 Data Recording

3.2.1 Data Classification

While we want to consider a more general approach to the topic of *web usage analysis* we want to give a overview on the types of data that can be found when we have the web as a potential source. The structure is based on [Srivastava *et al.*2000] and covers most of the aspects also relevant for the *active usage recording*. Special necessities for adapting this model are applied without further notice.

1. **Content:** The data which makes up the webpage (or the *state of the webpage*, when dynamic webpages are explored) the user is viewing. This data is mostly changing over time. Most often it can be influenced by the web application. Possible content types are

   - „simple" content, such as text and images and
   - multimedia content such as (interactive, animated) shockwave objects, music or video clips.

2. **Structure:** This term refers to data which describes the organization of the content. The main related scientific field of research is *semantic web*, where one goal is to make the web machine-readable.

   The structure within the *active usage recording* is heavily determined by the processes within the application. Thus the data recorded should be optimized with respect to these processes.

3. **Usage:** The data which describes the actual actions the users perform within the web application. This data is described in section 2.3.5 as *user feedback data*. The data model for it is presented in section 5.1.1.

4. **User Profile:** The user profile is very clear, since the application knows who is performing actions (per definition, see section 2.2.1).

3.2.2 Data from Active Usage Recording

Exemplary data from active usage recording is shown in table 2.7. The data used for the examples (e.g., Mosaicplot from figure 4.8) can be found in appendix C.

Example - Data from Active Usage Recording

Within the exemplary web application we can identify certain *actions* the users can perform. For the example given in this work, those actions might be

- BUY,
- RATE, and
- VIEW.

Moreover we have certain assets or goods which can be subject of those actions, such as

- ASSETS,
- RINGTONES, and
- ARTISTS.

Those actions, combined with a certain subject, occur on certain spots within the web application. This spots have an integrated trigger to write actions and their subjects into a database. Additionally *the point of time* when the action was performed is recorded.

The description of the exemplary web application an the description of the datamodel shows the important aspects of data from active usage recording:

- The data is related to the actions and processes the users perform.
- Subjects to the identified actions have to be recorded.
- Points of time when certain actions happen play a vital role for this kind of data.

3.2.3 Standard Server Log-File

A standard server log file structure is analyzed in [Haigh and Megarity1998] and [Srivastava *et al.*2000] and usually consists of the following entries:

- The **address** of the computer requesting the file,
- the **date and time** of the request,
- the **URL** for the file requested,
- the **protocol** used for the request,
- the **size** of the file requested,
- the **referring URL**, and
- the **browser** and **operating system** used by the requesting computer.

An example for a typical server logfile table is shown in table 3.2.

address (IP)	date and time	URL and protocol	size	referring URL	date and time
123.123.123.123	[26/Apr/2000:00:23:48 -0400]	"GET /pics/wpaper.gif HTTP/1.0" 200	6248	"http://www.-jafsoft.com/-asctortf/"	"Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123	[26/Apr/2000:00:23:48 -0400]	"GET /pics/5star2000.gif HTTP/1.0" 200	4005	"http://www.-jafsoft.com/-asctortf/"	"Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123	[26/Apr/2000:00:23:50 -0400]	"GET /pics/5star.gif HTTP/1.0" 200	1031	"http://www.-jafsoft.com/-asctortf/"	"Mozilla/4.05 (Macintosh; I; PPC)"
123.123.123.123	[26/Apr/2000:00:23:51 -0400]	"GET /pics/a2hlogo.jpg HTTP/1.0" 200	4282	"http://www.-jafsoft.com/-asctortf/"	"Mozilla/4.05 (Macintosh; I; PPC)"

Table 3.2: Standard Webserver Log-table

3.3 Web (Usage) Mining

The topic of *web usage mining* covers all the attempts of the DICE tool and the *web information cycle*. It is a general taxonomy for the techniques and products which deal with information and knowledge retrieval out of web based data generating processes (e.g., web applications) [Srivastava *et al.*2000]. Figure 3.1 is taken from [Srivastava *et al.*2000] and enlarged by contents provided in [Bolz2001]. It shows a general division of the major areas of web usage mining.

For this work, the major areas are all representing *views* on the information encapsulated in the data. The data pool is the same for all areas, in this case [Srivastava *et al.*2000] mostly addresses server log files as shown in section 3.2.3.

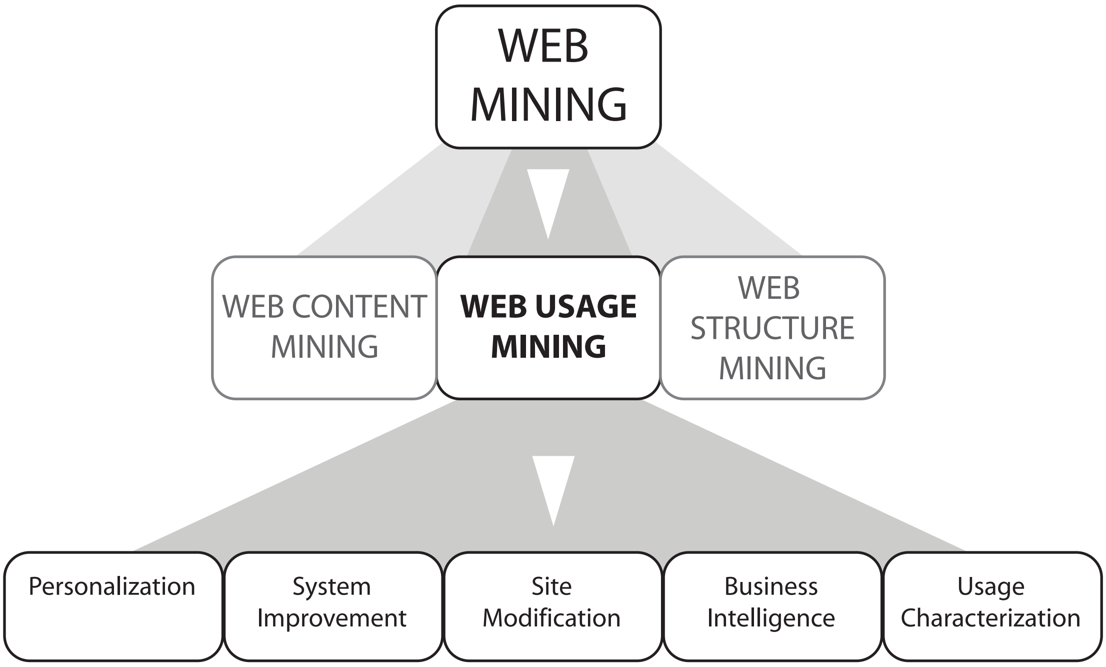


Figure 3.1: Major areas of web usage mining as presented by [Srivastava *et al.*2000] and [Bolz2001].

The approach of this work wants to widen the view on the topic of *web usage mining*. Essential enhancements should be

- the *active recording* of process oriented data,
- the usage of a consistent and compatible *data model*, and
- the use of *optimized tools* which apply useful visualization and interaction techniques.

To achieve this, one has to make choices from the very beginning. As addressed in [Srivastava *et al.*200 the *active usage recording* needs additional efforts in

- defining the necessary data,
- programming the web application,
- internet connection bandwidth,
- time while using the application, and
- providing incentives for achieving the necessary data.

3.4 Visualisation

3.4.1 Webserver Statistics

There are numerous products which aggregate the data from a standard server log file and represent it graphically. You can see typical representations in figure 3.2.

Implementations

The list of software products was chosen randomly with respect to representative visualization methods.

- **eTracker.de:** A platform which offers a great variety of services. The platform provides differentiated products for server log file analysis. The services range from realtime web statistics with *analysis of user campaigns* up to *live visitor tracking* [etracker GmbH2006]. Within the demo tour, we found interesting visualisation elements as follows:
  - **Calendar** for presenting the timely context of the data. The calendar is static and not animated (like we want to propose it within the time wheel in section 4.3.2). The existence of a calendar supports our proposition that the timely context of the data is very important for the analysis.
- **1-2-3 Log Analyzer:** A commercial product which uses a standard server log-file (see table 3.2 for an example) to perform basic statistical analysis. The visualizations contains the following [1] [ZY Computing2006]:
  - **Linecharts** of daily visitors.
  - **Barcharts** for the visualization of the aggregated user performance of weekdays and hours of the day.
  - **Tables** for presenting overviews of users origins or the most common search terms.

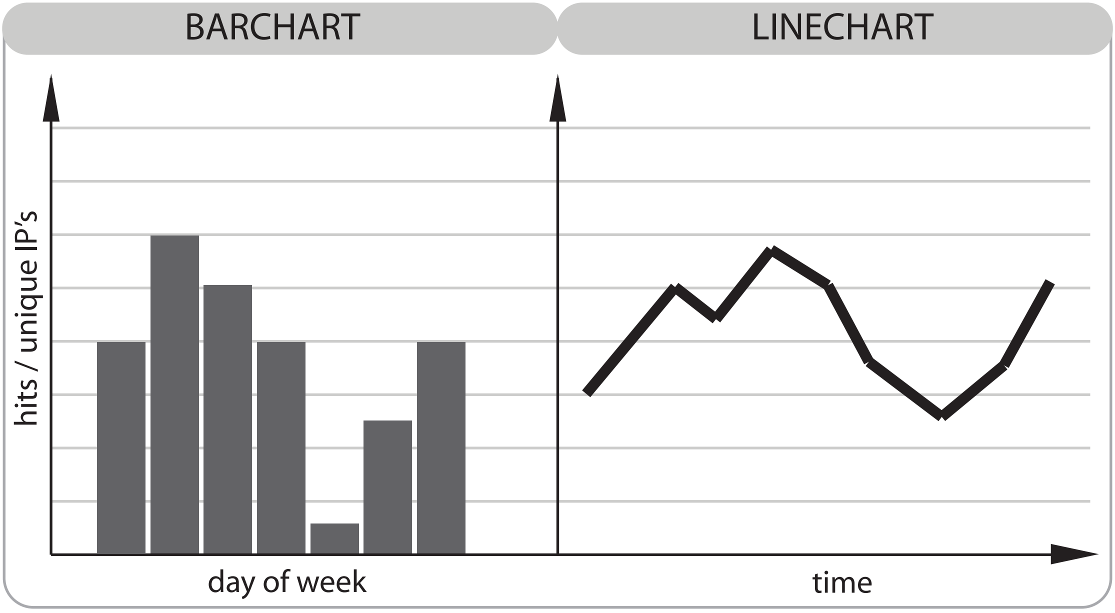


Figure 3.2: Standard webserver statistics offered in many implementations.

[1] Please note that the list of methods and examples is not exhaustive.

3.4.2 Graphs

One very basic definition of a website is the website as *a network of linked webpages* [Wikipedia2006e]. This goes back to the very beginning of the development of the World Wide Web in the early nineties. A very intuitive representation of a website is thus a representation through a graph. The nodes represent the single webpages and the edges represent connections between those pages which are the well known as *hyperlinks*.

Implementations

- **WebQuilt - Framework:** The framework introduced by Jason I. Hong and James A. Landay in 2001 extends the approach of static webpage networks and adds the timely component provided by most log file formats. Therefore, it offers a visualisation ,,... that shows the web pages traversed and paths taken" [Hong and Landay2001]. Webpages are represented as screenshots. The color of the edges shows the *length of the stay* on the webpage.

- **HostGraph:** The HostGraph is the very basic representation which is mentioned in [Feng *et al.*2006]. According to [Feng *et al.*2006], a HostGraph can be *naive* or *weighted*. See figure 3.3 (from [Feng *et al.*2006], page 2) for a representation.

- **Websites as Graphs:** The plain website `http://www.aharef.info/static/htmlgraph/` offers a very nice service. One can enter an URL and automatically receives a graph representation. The site itself is transformed with respect to certain tags. Those tags are represented with different colors. One can easily see the structure of the website [Sala2006].

3.5 Implementations

3.5.1 The Web Utilization Miner - WUM

The WUM project has now released three intermediate tools [WUMproject2006]:

1. The *DIAsDEM Workbench*,
2. *WUMprep* and its successor *WUMprep4Weka* and
3. the *web utilization miner (WUM)*.

We want to concentrate on the latter, the *web utilization miner*.

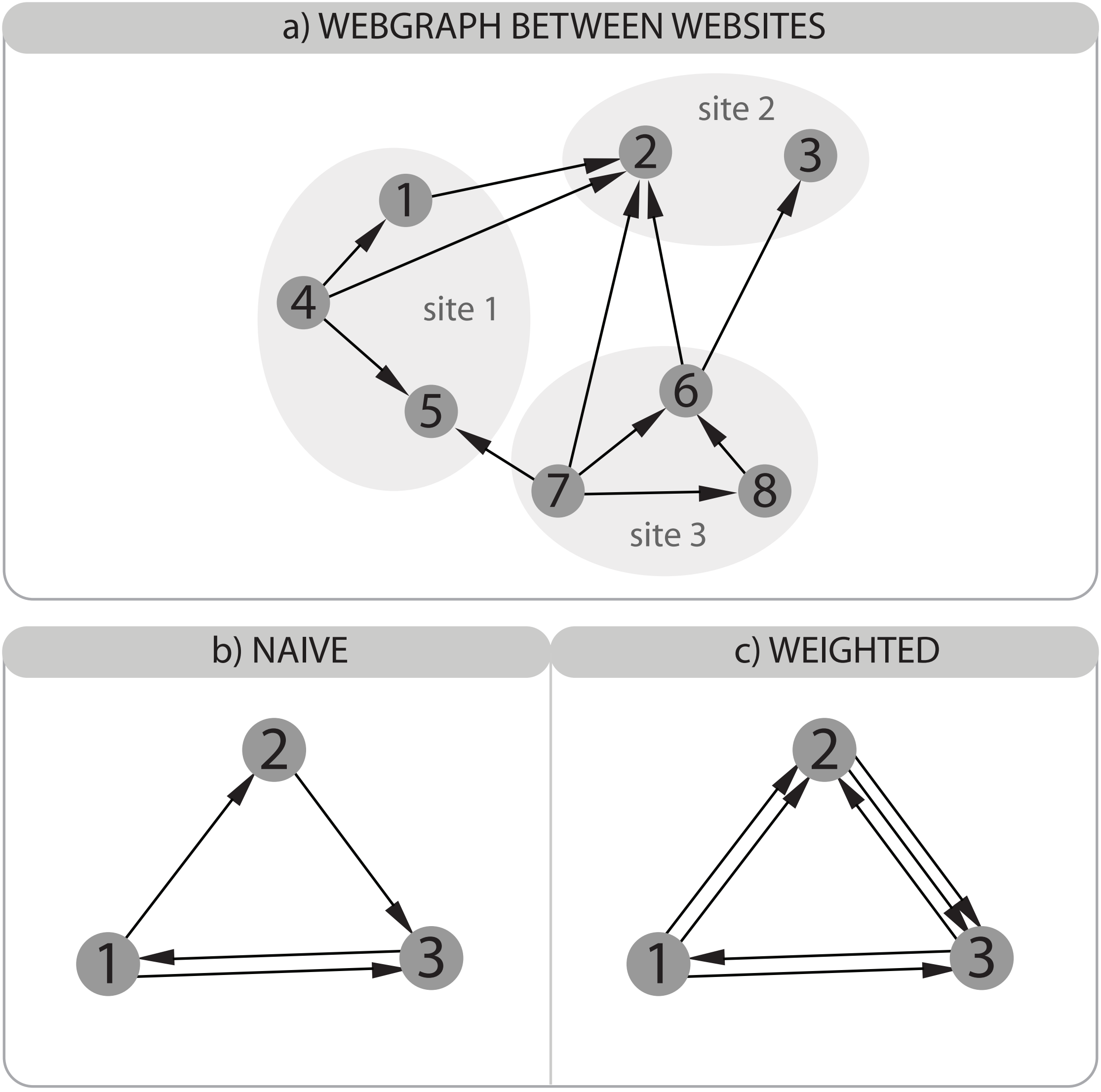


Figure 3.3: A model of a host graph showing the three relevant types.

The primary purpose of WUM is ,,... to analyse the navigational behaviour of users visiting a website." [WUMproject2006]. This statement implies, that the WUM is also suitable for a more general appliance. The key features of the WUM project are (cf. [WUMproject2006]):

- It is an **integrated environment**.
- The analyst can use functions for the **preparation of log file data**.
- It has its own **query language (MINT)** for querying log files.
- The WUM can be used for **general type of log files**.
- An so called ,,aggregation service" is part of the WUM for generating a tree structure of the analysed log data.
- It uses **visualisation** to present results of the analysis.

The general approach of the WUM to fit more than one log data type is very interesting. Being able to connect a data source to the analysis tool is also part of the DICE implementation.

The usage of a query language sounds like a powerful tool for analysts. Nevertheless, the DICE approach wants to enable the tool to provide only visual controls. As we know, text based input is needed to provide consistent controls in the first place. Maybe future versions of WUM will provide graphical interfaces for the MINT language.

3.5.2 Sawmill

A few of the key areas of the Sawmill software are [Flowerfire2006]

- web server analysis,
- mail server analysis, and
- security monitoring and management.

Key aspects, which can be related to the work presented in this thesis, are (cf. [Flowerfire2006]):

- The generated report can be **drilled down** via zoom filters.
- An interface to the well known **mySQL database** is provided.
- Internal control via a special language interface called *Salang*.
- The analyst can perform **data preparation methods** on the raw data.
- Support for 635 different log file formats.
- It is possible to **integrate more than one data source** for the analysis.

3.5.3 DBMiner

The DB Miner project has ,, ... developed major innovations in data mining techniques and algorithms." [DBMiner2006].

Key aspects, which can be related to the work presented in this thesis, are (cf. [DBMiner2006]):

- DB Miner provides **openness** for various data sources.
- The tool is able to work **on-line** and perform analysis directly on the database.
- The project offers **advanced visualization** functions.
- Analysts find well integrated **statistical concepts** within the DB Miner tool.

The DB Miner set of tools are divided into several products (or bundles) which are equiped with he following features (cf. [DBMiner2006]):

- Answering of less-well-defined questions which have little technical relevance but high business importance (exemplary questions are from [DBMiner2006]).
  - Which customers prefer product A over product B?
  - When are these customers likely to buy additional products or services?

  - What financial transactions are suspicious?
  - What products in which location and at which month have generated significant profit increase?

- The DB Miner uses **intelligent** and **automated processes**.
- One of the goals of DB Miner is to **generate knowledge** out of high amounts of data.

The products offered by *DB Miner Technology Inc.* sound very interesting and highly professional.

As stated by [DBMiner2006] it follows very much the long-term goal of the DICE approach: ,,... you can present your products or services to customers in a way that's most likely to increase your value to them and their value to you. ". From the information we could gather about the products of DB Miner Technology Inc. we think the software has a lot in common with our concept.

3.5.4 STstat

The STstat project is also dealing with the analysis of HTTP log files. It is a tool for analysing the traffic on a website based on the data of the webserver.

From [Software2006] we derived the following key-aspects fo the STstat tool:

- The analyst can manage **profiles** for the analysis of more websites.
- The analyst has access to **reports** which show aggregated views on
  - remote visitors,
  - local files/URLs,
  - remote domains,
  - remote countries,
  - referrers,
  - remote user agents,
  - authenticated users, or
  - banner advertisements.
- The tool uses **zooming** (see section 4.1.2 for details on zooming) for a deeper insight into the generated reports.
- Traffic statistics can be **wrapped** over a certain timespan such as
  - day of week,
  - hour of day, or
  - single day.
- Moreover, STstat works **real-time**, that means you can process the log files from the server while the database is still up and running.

The tool seems to be a very interesting reporting tool for webserver or website administrators. Nevertheless, we identify the following aspects which are insufficient for the DICE - approach.

- The STstat tool does not support *active usage recording* as proposed in this work.
- There is no explicit possibility of interaction with the views on the data.
- Statistical methods are used on a very basic level.
- The timely aspect of the data is integrated insufficiently.
- There is no process view of the log data.

3.6 Conclusion

The chapter presented work in the field of web mining and web statistics. It is undoubtedly a very large area, where a lot of implementations make a promising impression. For a lot of implementations, the root seems to be standard log file analysis. The investigation showed a lot of focus on log file analysis and knowledge derived from this type of data. The analysis implementations are normally decoupled from the data generating web application. This contradicts to a certain extent with the DICE approach, where the active recording of usage data is said to be a substantial part of the implementation.

Chapter 4

Methods

> ❞ He that would perfect his work must first sharpen his tools. ❝
> - Confucius -

The use of *visualization methods* is important when we work with large data sets and (relatively) little space to view it. This leads to techniques which perform *data aggregation*. They provide *selective data browsing* through the data space [Leung and Apperley1994]. Especially when we have little or no knowledge about the patterns within the data, we need visual techniques to explore the data. The visual *data exploring process* can be seen as a hypothesis-generation process where the creation of hypothesis is done iteratively [Keim2001]. This is consistent with the *information seeking mantra* introduced by [Shneiderman1996]. There we also have an iterative approach to data exploration with visual methods.

The following sections want to give a clear understanding of the abilities of the tool. We will divide the abilities into three groups:

- Interaction techniques,
- visual methods with a statistical focus, and
- visual methods.

Each group of abilities will be described and justified. Then we will provide a detailed description of the abilities. For that we use a graphic which pictures the scale of the ability in two dimensions. The first dimension, *DATA TYPE*, shows which types of variable can be handled by the function. The second dimension, *INTERACTION AND MORPHING TECHNIQUES*, shows which interaction and morphing techniques can be applied to the function.

The figures (4.7, 4.9, 4.11 and 4.14 as well as the general overview figure 4.1) are all based on the *classification of visual data exploration techniques* figure by [Keim2001].

Moreover, we want to clarify how the described visual method or interaction technique will cooperate with integrated parts of the tool (such as the timeline). We also present an example for each ability.

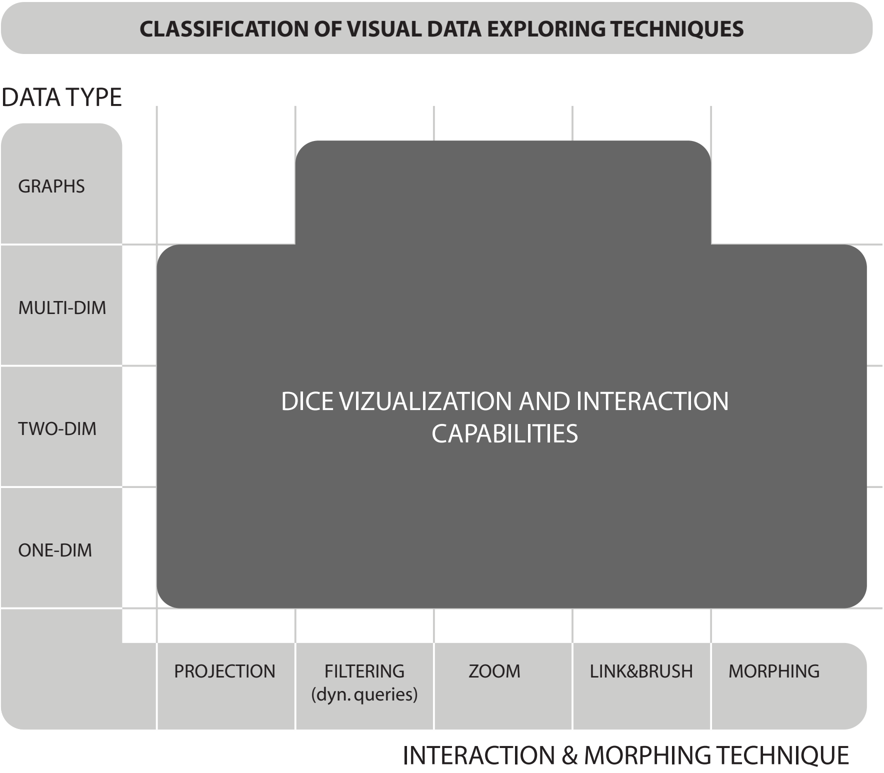

CLASSIFICATION OF VISUAL DATA EXPLORING TECHNIQUES
DATA TYPE
GRAPHS
MULTI-DIM
TWO-DIM
ONE-DIM
DICE VIZUALIZATION AND INTERACTION CAPABILITIES
PROJECTION
FILTERING (dyn. queries)
ZOOM
LINK&BRUSH
MORPHING
INTERACTION & MORPHING TECHNIQUE

Figure 4.1: The DICE outline within the classification scheme of visual methods.

4.1 Interaction Techniques

Interactivity within visual exploration methods has several advantages over automatic and non-interactive data mining or statistical techniques [Keim2001]:

- Data errors (missing values, inequalities, bias) do not disturb the analysation process to a significant extent.
- The process itself is under the analyst's control, thus stages can be defined (compared to a ,,run through" routine in some statistical or data mining tools).
- The need for understanding the complex structure of the underlying data and the data generating process is very much reduced.

Our interaction techniques follow the definition in [Hofmann2000], where interaction techniques are said to be *potentially interactive*, when they aim to be as fast as possible. But there are no timeframes defined in which the operations must take place to be considered interactive.

[Hofmann2000] points out the need for interaction techniques in visualization methods for the representation of *categorical data*. This conforms to the proposed data model (cf. 5.1.1) for the DICE tool, where the variables *USER ID*, *ACTION ID* and *SUCCESS* are factors.

4.1.1 Dynamic Queries

The concept of dynamic queries was introduced in [Shneiderman1994], [Williamson and Shneiderman1992] and [Ahlberg *et al.*1992]. The following definition is given in [Shneiderman1994]:

Definition - Dynamic Queries

,, *Dynamic queries continuously update the data that is filtered from the database and visualized. They work instantly within a few milliseconds as users adjusts sliders or select buttons to form simple queries or to find patterns or exceptions; the dynamicquery approach thus applies the principles of direct manipulation to the database. [Shneiderman1994]* ''

[Ahlberg *et al.*1992] has proven that the concept of *dynamic queries* can be successfully enhance the usability and reduce the expenditure of time of a user interface.We want to avail the results for the DICE tool.

Within the tool we can apply the concept of *dynamic queries* to the following elements:

1. **Timeline:** The active timespan can be selected within the *time line.* The relevant starting point and end point can be set with a slider. The span inbetween selects the data which is used to generate the figure. This means that a person can dynamically select a special timespan in the data and look at it more closely. Figure 4.4 presents the buttons and sliders of the timeline in detail. You might want to look at section 4.1.3 and figure 4.4 for further details.

Example - Dynamic Queries with Timeline

The DICE tool is successfully connected to a data source. All of this data has a *sorted timestamp*, following our proposed datamodel (see 5.1.1). Thus we can select a subset of the data with respect to a certain time interval. This is done by using the timeline.

In figure 4.2 we see how the selection of a time interval affects the sequence of figures derived from the subset (with respect to the selected aggregation level).

We want to point out the necessary aspects of an interface to include dynamic query functionality (according to [Williamson and Shneiderman1992]):

- **Graphical representation** of the request: The design of the timeline (as shown in figure 4.4) follows given implementations from [Ahlberg *et al.*1992] and [Williamson and Shneiderman1992]. Therefore, it is considered to have an adequate graphical representation of the request.
- The graphical **visualisation of the database and searching results**.
- Delivers **results immediately** when several parameters are changed: A change in the selected time interval through the timeline object is immediately affecting all other representations of the data (see 4.2 for a representation).
- **Visualizes result borders** (min-max): The borders of the maximal selectable range is shown either as relative amount (in percent) or as absolute timestamps (see figure 4.4).
- **Allows beginners a faster entrance** without having much practice, and still **offers experts some mighty functions**:
  - Beginners will find the selection of start and end point very intuitive.
  - It is also clear that the selected timespan queries the data, because the other representation adopt instantly.
  - Experts can also shift the whole interval at once, changing the start and end point with constant interval length.
  - With some experience the analyst can combine the multiview application with the timeline for extra insight into the data.

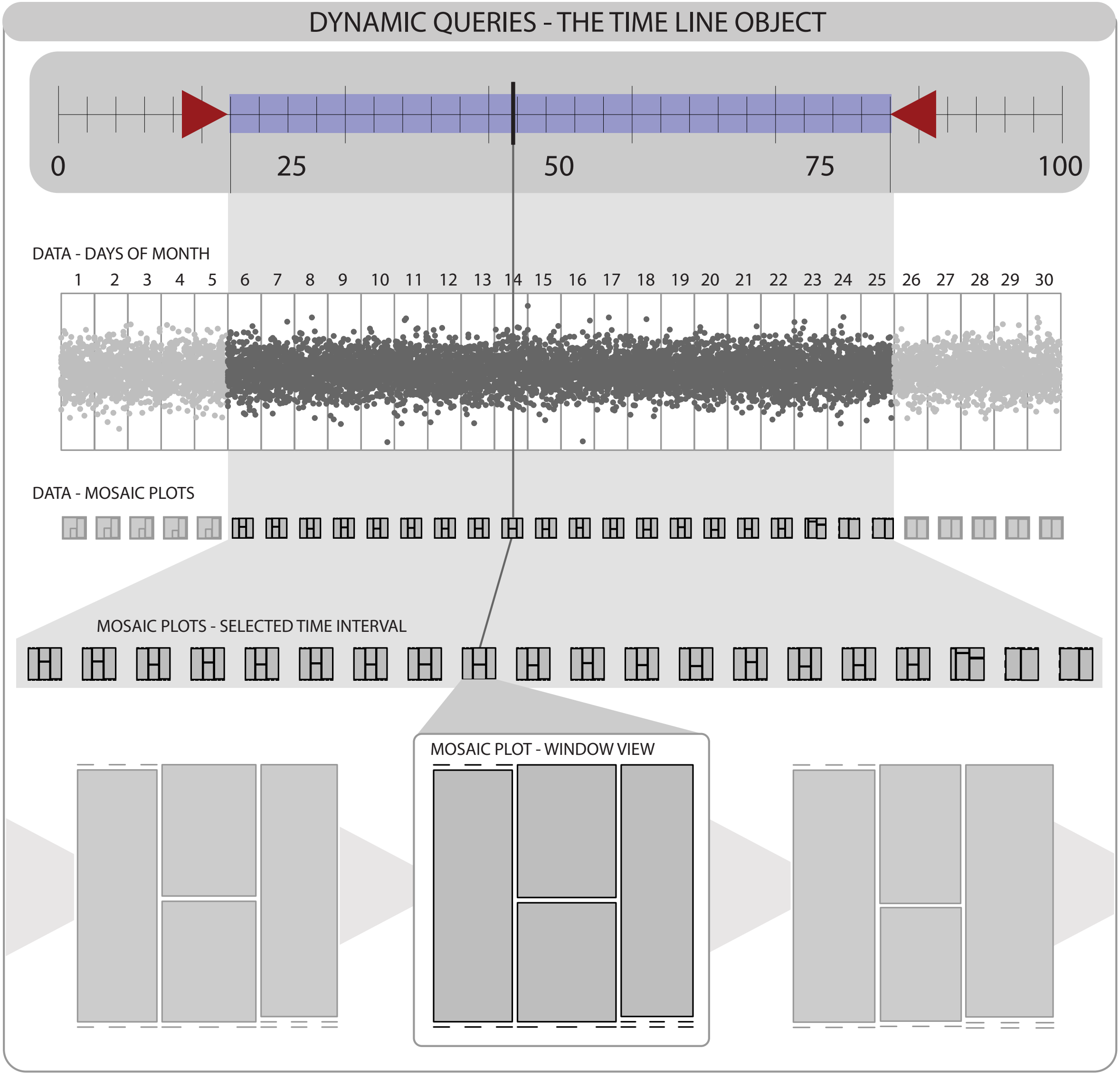


Figure 4.2: The subset selection via the timeline represented in different levels. The first and second level shows the data selection by the timeline, the other levels present the selection of the aggregated figures (e.g., the mosaicplot).

4.1.2 Zoom

We want the user to be able to perform the zooming as an *integrated procedure*. This means that the user does not have to query the whole figure again. The user should achieve the results iteratively and should be able to zoom in and out from a previous result.

In this case zooming happens when we change the aggregation level of a variable. This also selects a subset of the data and thus reduces the data in use. Because the selection of a subset can significantly influence the representation of the data (e.g., within the mosaicplot), we can also speak of *semantic zooming* [Boulos2003].

Definition - Semantic Zoom

,, *With a conventional geometric zoom all objects change only their size; with semantic zoom they can additionally change shape, details (not merely size of existing details) or, indeed, their very presence in the display, with objects appearing/disappearing according to the context of the map at hand. [Boulos2003]* ”

Example - Zooming by Aggregation Level

The example wants to show that the change of aggregation level can perform a zoom and in addition, change the look of the figure to a significant extent.
If we look at figure 4.3 we see two figures representing jittered dotplots (or jitterplots). The figures have a different aggregation level. The figure on the left side shows data with respect *to the hourly occurrence* of the action. The figure on the right hand side, on contrary, shows the same data with respect *to the daily occurrence* (in a weeks time).
One can see significant difference in the structure of the data. Thus it is important to be able to change attributes of the visualisation methods.

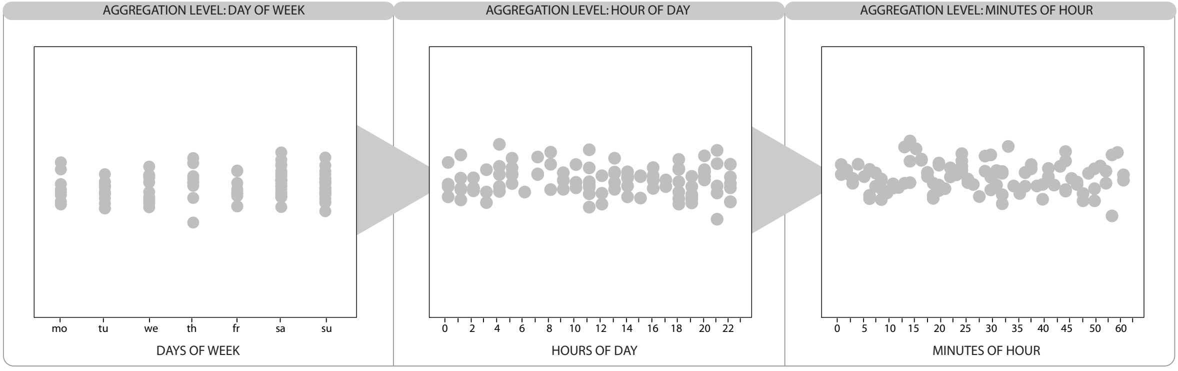


Figure 4.3: Jitterplots with different aggregation levels can uncover differences in structure or distribution of data points.

4.1.3 Timeline

The data model for usage data for the DICE tool has an obligatory time stamp. This means that analysis of the data over time is an inherent part of the tool. Data without time stamp will only work as additional information source within the DICE tool.

The interaction with that time stamps has certain possibilities. One of them is the timeline. The timeline has to be interactive. This means that selecting different time regions has to be easy and intuitive. Thus, there has to be a graphical interface which gives the opportunity to select different time spans. We have already presented the details of the interaction capabilities in section 4.1.1. We have also shown how the timeline is connected to the data and the figures. We want to go on with presenting the realization of the timeline.

One possible technique to select time (and thus value) regions dynamically by a graphical interface was introduced as *time boxes* by [Hochheiser and Shneiderman2001]. This idea was enlarged by [Keogh *et al.*2002] under the name of *variable time boxes*. Variable time boxes are a most general case of the *timeline* implemented in the DICE tool.

The specialisations compared to variable time boxes are:

- There is only **one pair of sliders** to set the region of interest **for one variable**.
- Every variable suitable for dynamic querying is set by a different pair of sliders.
- The interesting **timespan can be set independently** and is not connected to any other parameter.

Definition - Timeline

,, *The* timeline *in the DICE implementation is a single, variable Time Box, where the constraints are set with independent control instances.* ”

The *timeline* implementation in the DICE tool will consist of the following parts (see figure 4.4 for a possible composition):

- Relative or absolute **axis description**.
- Two **buttons to select the left and right margin** of the timespan to be observed.
- A bar which shows the **current selection of time**.
- A **playhead**, representing the current point of time shown on screen.
- **Control buttons** for the playhead with a metaphor to a music player:
  - PLAY, STOP and PAUSE buttons for control of the playback.
  - A speed control slider representing the speed with which the playhead moves forward.

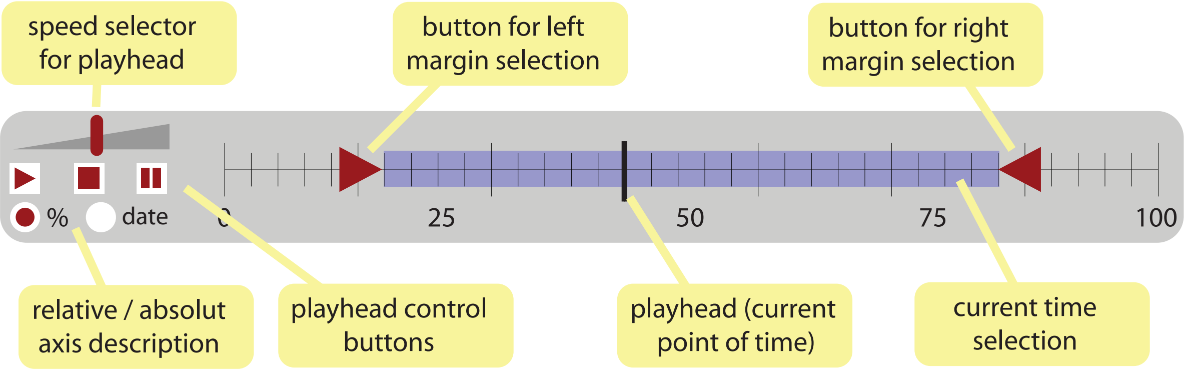

speed selector for playhead
button for left margin selection
button for right margin selection
% date
0
25
50
75
100
relative / absolut axis description
playhead control buttons
playhead (current point of time)
current time selection

Figure 4.4: The timeline has several tools which can be used interactively to select different time intervals of relevant data. The interval itself can also be shifted via the drag-and-drop technique.

4.1.4 Linking and Brushing

The workspace of the tool will be capable of presenting more than one figure at once. This provides a wider view on the problem.

A good definition of the linkage between multiple views can be found in [NIST/SEMATECH2006]. It defines linking as follows.

Definition - Linking and Brushing

,, *By linking, we mean showing how a point, or set of points, behaves in each of the plots. This is accomplished by highlighting these points in some fashion. For example, the highlighted points could be drawn as a filled circle while the remaining points could be drawn as unfilled circles. A typical application of this would be to show how an outlier shows up in each of the individual pairwise plots. Brushing extends this concept a bit further. In brushing, the points to be highlighted are interactively selected by a mouse and the scatterplot matrix is dynamically updated (ideally in real time). That is, we can select a rectangular region of points in one plot and see how those points are reflected in the other plots. [NIST/SEMATECH2006]* ”

Example - Linking and Brushing: Mosaicplot and Jitterplot

A good example for *a linkage* between the figures is the time. The variable *time* is present in every visual method. We can use the timeline to set the actual timespan. All figures will refresh instantly. The refreshed figures will only show the data which occurred within the selected timespan (please see figure 4.2 for a representation). This is called *the link between figures.*

We can also use other figures to select data points. The tool will provide a *brush*, which can be used to select data points inside a figure. These data points will be highlighted in the figure *and in linked figures.* Of course the data points will occur within a different place. This might reveal hidden connections or clusters in the data.

The process of selecting regions of the figure with a tool (e.g., a brush) is called *brushing* [NIST/SEMATECH2006]. An example for the effect of *linking and brushing* is shown in figure 4.5. The red region which was ,,brushed” in the mosaicplot appears in the figure on the right as red dots. With this technique the analyst can visualize how subsets of action times are distributed.

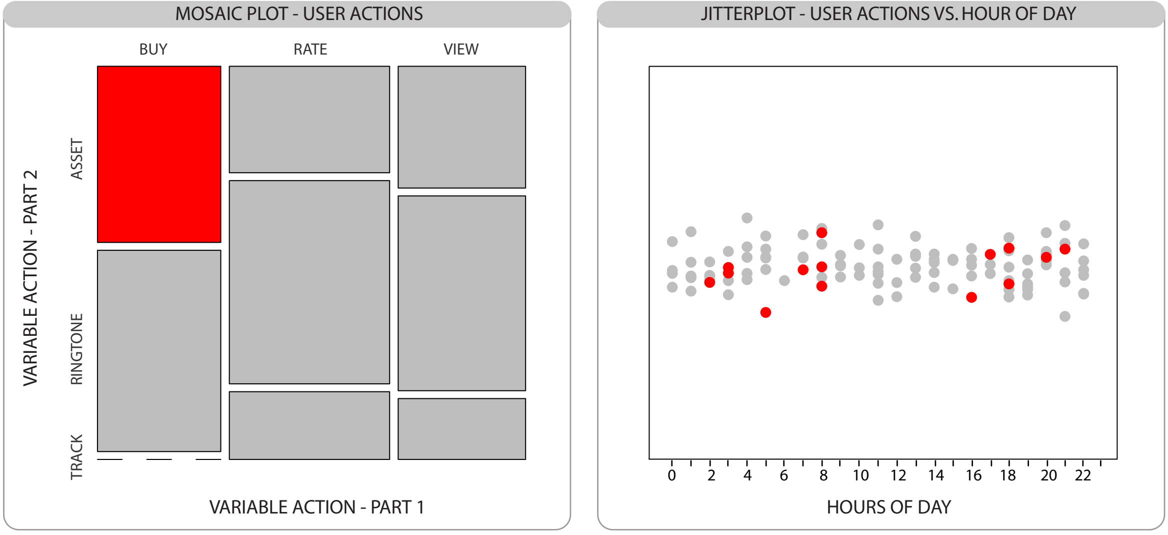


Figure 4.5: The linkage between figures can reveal the position of data points over different views. In our example a subset filtered by two factor levels (ASSET and BUY) is enhanced by the relevant point of time. This is done by selecting the points in the mosaicplot. The jitterplot marks the selected data points.

4.1.5 Morphing

Figures like the mosaicplot cannot be displayed as an animation over time. A new figure will always overplot the last figure. Because of that we have *abrupt transitions* in the flow of figures. These transitions are very interesting, because they give insight into the development of the data over time. To be able to get insight into the change which happen at the transition of the figures, we need to slow down those transitions. The idea is to make a visual fading of the figures. The new figure will *fade in*, and at the same time the old figure will *fade out*. This will provide a timespan where the transition of the single elements will be seen clearly. We have provided a series of images in figure 4.6 which clarify the *morphing technique*. In this figure we present the transition between two mosaicplots. Every plot is generated from a certain time interval. The transition should run smooth to be able to recognize significant changes in the figures. When the transition process starts, figure one (the old figure) has full opacity whereas figure two (the new figure), has no opacity at all. As time passes, the opacity level of figure one falls and the opacity level of figure two rises. This means, that the visible figure will change smoothly from figure one to figure two. The timespan this *transition procedure* lasts depends on the animation speed. This speed can be set with the controls of the timeline (cf. 4.1.3).

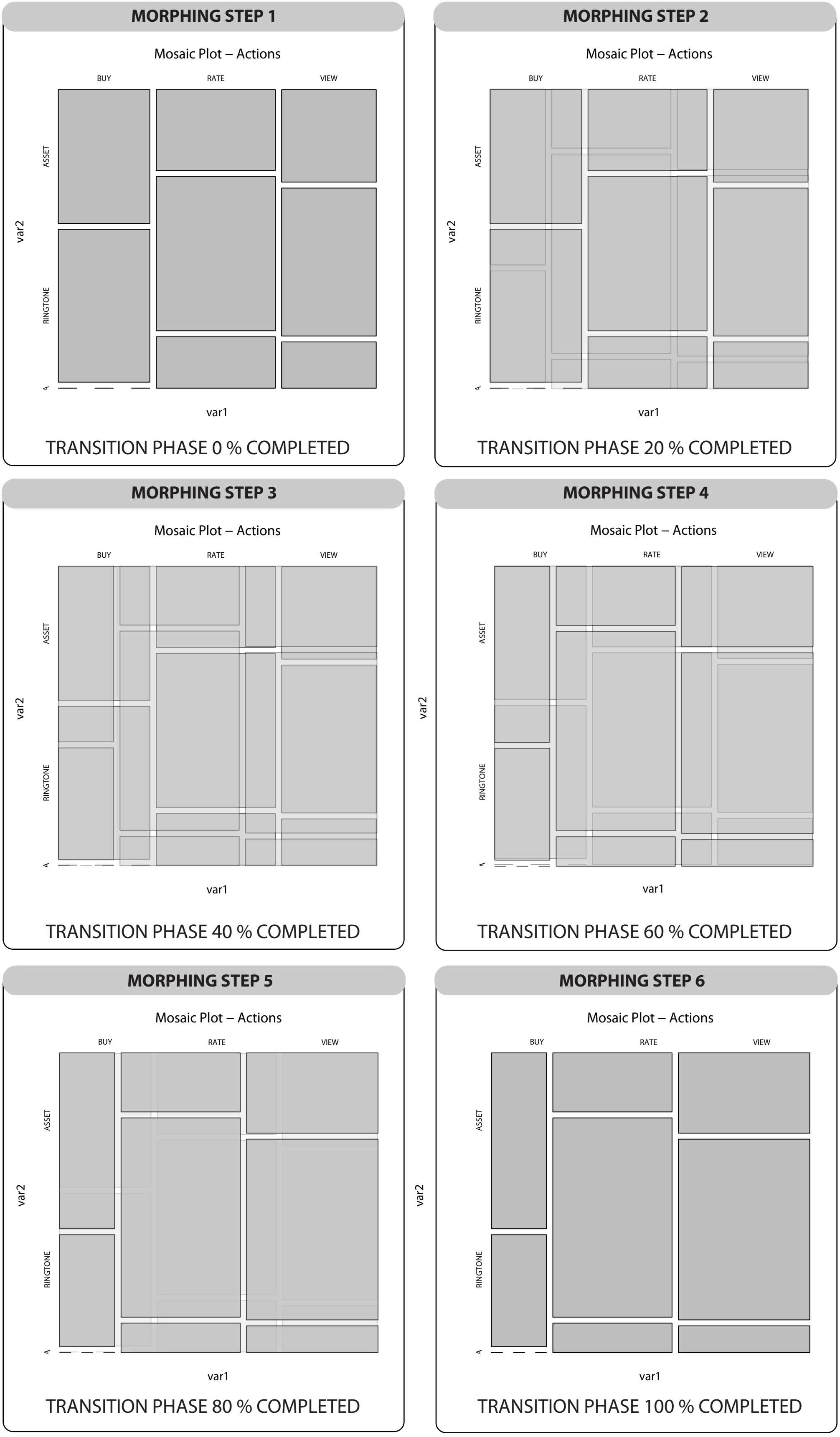


Figure 4.6: Six positions of the mosaicplot while morphing from one representation to another.

Example - Morphing of the Mosaicplot

In figure 4.6 we can see the transition between two exemplary mosaicplots. The first mosaicplot (which can be seen with full opacity in the top left corner) is based on the last time interval. The second plot (which can be found in the lower right corner) is based on the coming time interval. The data was randomly generated. We now want to have a closer look at the transition.

- In **step 1** we see the figure based on the last time interval. Let us focus on the x axis. The bar width divides the axis with respect to the amount of BUY, RATE and VIEW actions in the data.
- **Step 2** already shows the new figure with 20% opacity. The opacity of figure one is reduced to 80%. Figure two has a thinner bar for the BUY action. This means, that fewer BUY actions were performed in the coming time interval.
- In **step 3** there is not much of a change, we see the picture *melt* into one another.
- **Step 4** and **step 5** give a clear hint that the next time interval has less BUY actions than the one before.
- **Step 6** finally presents the figure for the next time interval in full opacity. The old figure has disappeared.

The analysis done for the relative proportion of the BUY actions can be performed in a similar way for all the other factor levels in the plot.

4.2 Visual Methods with a Statistical Focus

During the data exploring process we create hypothesis. Further proof can be achieved with methods, like statistical tests. There are some visual methods which combine the benefits of visual and statistical methods. The *mosaicplot* for instance provides a clue for the significance of the deviation of the proportions within groups.

This methods *cannot* replace statistical tests. But they can enhance the visual process to be more evident. As a result we can verify and preselect the hypothesis generated [Keim2001].

Example - Using Statistical Methods: Mosaicplot

The use of statistical methods is especially important to visualize *statistical significance*. The contingency table 4.7 shows the absolute counts of exemplary user actions.

Our interest is wether the type of the good (ASSET, TRACK, RINGTONE) has an impact on the action performed. We can write the hypothesis of this *multinomial problem* as follows [Christensen1990]:

$$H_0 : p_{i1} = p_{i2} = p_{i3} \tag{4.1}$$

In other words, we test for the *homogeneity of proportions* [Christensen1990]. Each *column* of table 4.7 is a binomial with sample size $n_{.j}$. This leads to the natural estimate of the cell probability [Christensen1990]:

$$\hat{p}_{ij} = \frac{n_{ij}}{n_{.j}} \tag{4.2}$$

We compute the Pearson's chi-squared test statistic as follows:

$$X^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{\left(n_{ij} - \hat{m}_{ij}^{(0)}\right)}{\hat{m}_{ij}^{(0)}} \tag{4.3}$$

Given the null hypothesis is true (and thus the model in 4.1 is true), we can estimate m_{ij} with (cf. [Christensen1990]):

$$\begin{aligned} \hat{m}_{ij}^{(0)} &= n_{..}\hat{p}_{i.}\hat{p}_{.j} && (4.4)\\ &= n_{..}(\frac{n_{i.}}{n_{..}})(\frac{n_{.j}}{n_{..}}) && (4.5)\\ &= \frac{n_{i.}n_{.j}}{n_{..}} && (4.6) \end{aligned}$$

The estimates for the $\hat{m}_{ij}^{(0)}$ values can be found in table 4.8 and table 4.9. If the sample size is large the X^2 statistic is approximately $\chi^2_{((I-1)(J-1))}$ distributed. The null hypothesis (see 4.1) is rejected at the α level, if

$$X^2 > \chi^2_{(1-\alpha,(I-1)(J-1))} \tag{4.7}$$

For the exemplary data we receive $X^2 = 6.9554$ as test statistic which leads to a p-value of 0.1383. This means, that given the null hypothesis from model 4.1 is true, we can observe more unlikely data (in terms of distance from the null hypothesis) with probability of 0.1383 (cf. [Wikipedia2006b] and [Christensen1990]).
At a reasonable confidence level (e.g $\alpha = 0.05$), the null hypothesis cannot be rejected.

4.2.1 Mosaicplot

The mosaicplot, attributed to Hartigan and Kleiner (1981), is a graphical method to display the cell frequencies of contingency tables as a plot. It has become a primary tool for investigating categorical data in the form of n-way contingency tables [Friendly2001].

Figure 4.7 gives an overview how our mosaicplot implementation fits into the space of data type and interaction possibility.

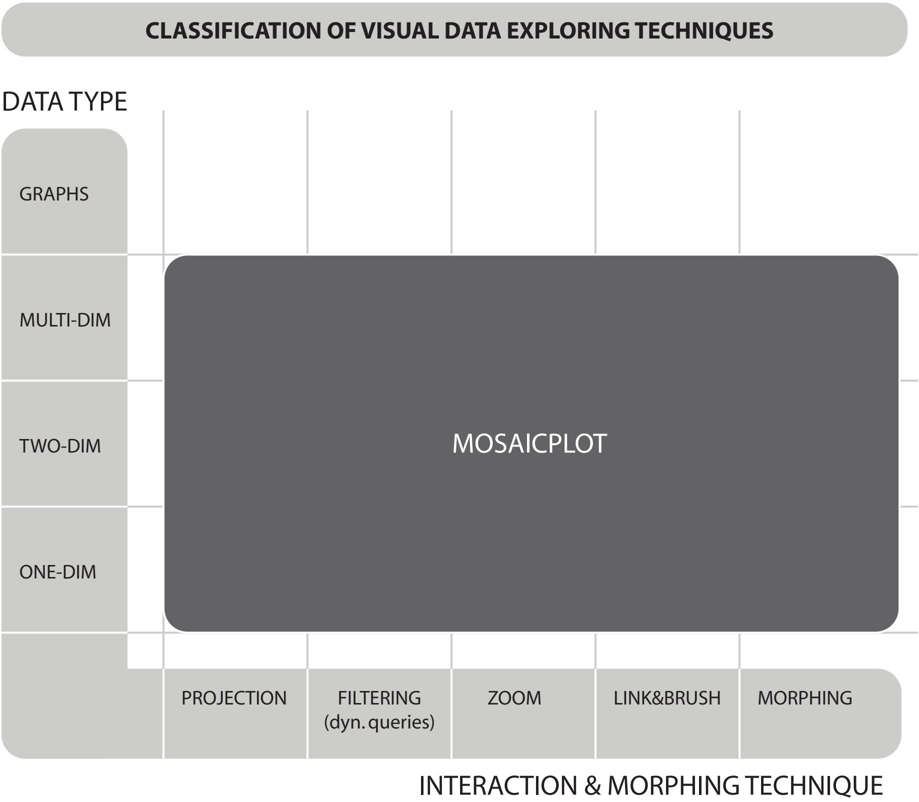


Figure 4.7: The possible features of the mosaicplot within the classification scheme.

The graphical features of the mosaicplot, following [Friendly2001], are:

- One plot is divided into *bars*, which represent one variable.
- The width of the bars represents the marginal totals $n_{i.}$ (of course this is also proportional to the marginal probabilities $p_{i.} = \frac{n_{i.}}{n_{..}}$).
- Each bar is divided into *tiles*, whose heights represent the actual cellcounts n_{ij}.
- The area of a tile represents the *cell frequency* n_{ij} in the cross classification of the variables.

Algorithm

The algorithm for the mosaicplot is implemented in statistical packages like S-PLUS and R. Therefor an efficient recursive procedure, which can handle many factors, exists [Emerson1998][1].

Example - Mosaicplot

An example mosaicplot with hundred simulated records is shown in figure 4.8. One can easily see the distribution of the cell counts which is very clearly represented by the filled areas. The underlying contingency table can be seen in table 4.7.

[1] For further information and a detailed description of the algorithm you might want to look at [Emerson1998] and [Friendly2001]

Hypothesis derived from this plot could look as follows:

- Customers cannot BUY TRACKS.
- There is no significant difference in the amount of RATE and VIEW actions.
- Customers who RATEd a good tend to BUY it more likely than those who did not RATE the good.

The hypothesis can now be verified using Chi-Square test statistics which measure the independence of the factors [Christensen1990].

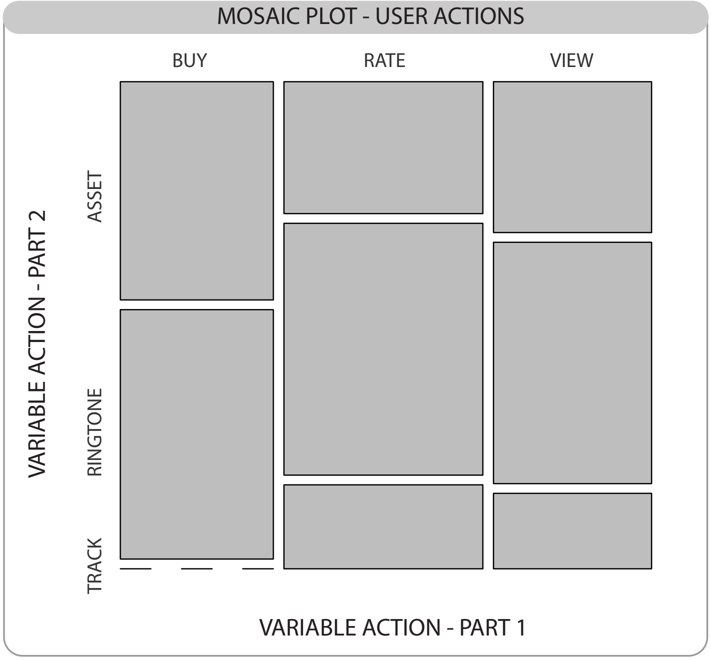


Figure 4.8: A mosaicplot of simulated usage data comparing users actions and related goods.

	ASSET	RINGTONE	TRACK	
BUY	14	16	0	**30**
RATE	11	21	7	**39**
VIEW	10	16	5	**31**
	35	**53**	**12**	**100**

Table 4.7: A contingency table presenting the variables USER ACTIONS versus GOODS.

	ASSET	RINGTONE	TRACK	
BUY	$\frac{35\times30}{100}$	$\frac{53\times30}{100}$	$\frac{12\times30}{100}$	**30**
RATE	$\frac{35\times39}{100}$	$\frac{53\times39}{100}$	$\frac{12\times39}{100}$	**39**
VIEW	$\frac{35\times31}{100}$	$\frac{53\times31}{100}$	$\frac{12\times31}{100}$	**31**
	35	**53**	**12**	**100**

Table 4.8: The table of $\hat{m}_{ij}^{(0)}$ estimates.

	ASSET	RINGTONE	TRACK	
BUY	10.50	15.90	3.60	**30**
RATE	13.65	20.67	4.68	**39**
VIEW	10.85	16.43	3.72	**31**
	35	**53**	**12**	**100**

Table 4.9: The table of $\hat{m}_{ij}^{(0)}$ estimates.

4.2.2 Wrapped Jitterplot

The jitterplot is a very simple statistical method. The goal is ,,... to give a clue about possible differences in the density of some data.” [Wolf1999]. When we use data aggregates to produce statistics (such as a histogram), we have to reduce the given information. In case of a histogram we would represent data point as areas.

Within the DICE tool we have a lot of categorical data. This data has one continuous variable, the timestamp. So if we plot *actions over time*, we will get a black line because of heavy overlapping of the data points (see the left part of figure 4.10 for a representation). Deriving information of the *occurrence frequency* of actions is almost impossible.

To get a clue about the frequency of the data we randomly jitter it around the y axis. By doing so we receive a picture which is giving insight into the structure of the occurrence of the data. The main disadvantages of the methods are the ambiguity of the figure and the dependence on the chance when the figure is generated.

$$y(d) = d + N_{(\mu,\sigma)} \tag{4.8}$$

Equation 4.8 shows the computation of the y values $y(d)$ of the plotted points. d is a constant, most often zero, to which we add a normally distributed value with mean μ and standard deviation σ. In the examples we used $\mu = 0$ and $\sigma = 0.1$.

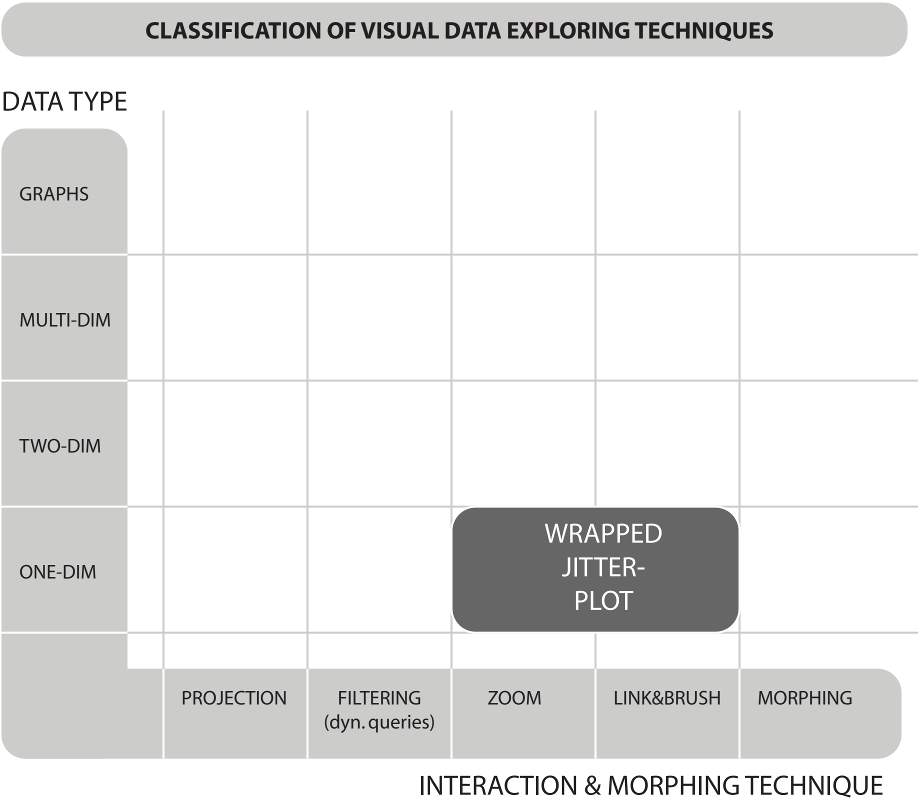


Figure 4.9: The possible features of the wrapped jitterplot within the classification scheme.

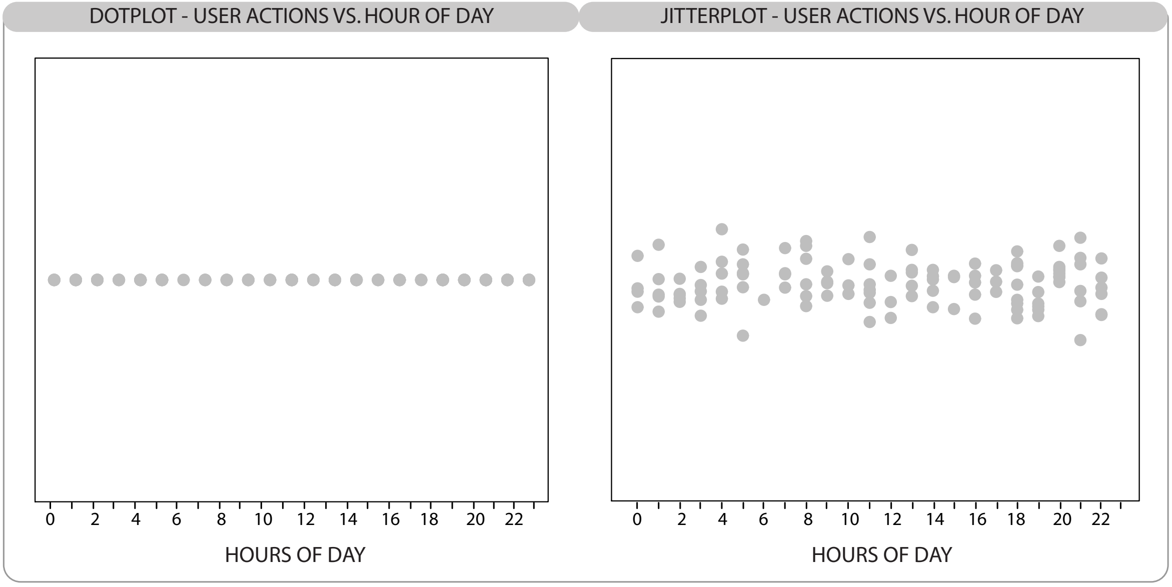


Figure 4.10: An example of a wrapped jitterplot generated from simulated usage data showing the difference of a regular dotplot compared to the jitterplot. The jitterplot gives an idea about the distribution of the data over time.

4.3 Visual Methods

4.3.1 Logsnakes

Logsnakes were a special development for a defined task. The task can be generalized in the following terms:

- Customers have accounts.
- Operators want to bill those accounts.
- Operators can make attempts to bill accounts and get responses. Those can be either *success* or *failure*.
- According to the response, they follow a certain strategy.

There were some goals defined for the technique:

- Get a general idea about the data structure.
- Focus on response code.
- Visualize the strategy of the different operators.
- Maybe find regions of interest.

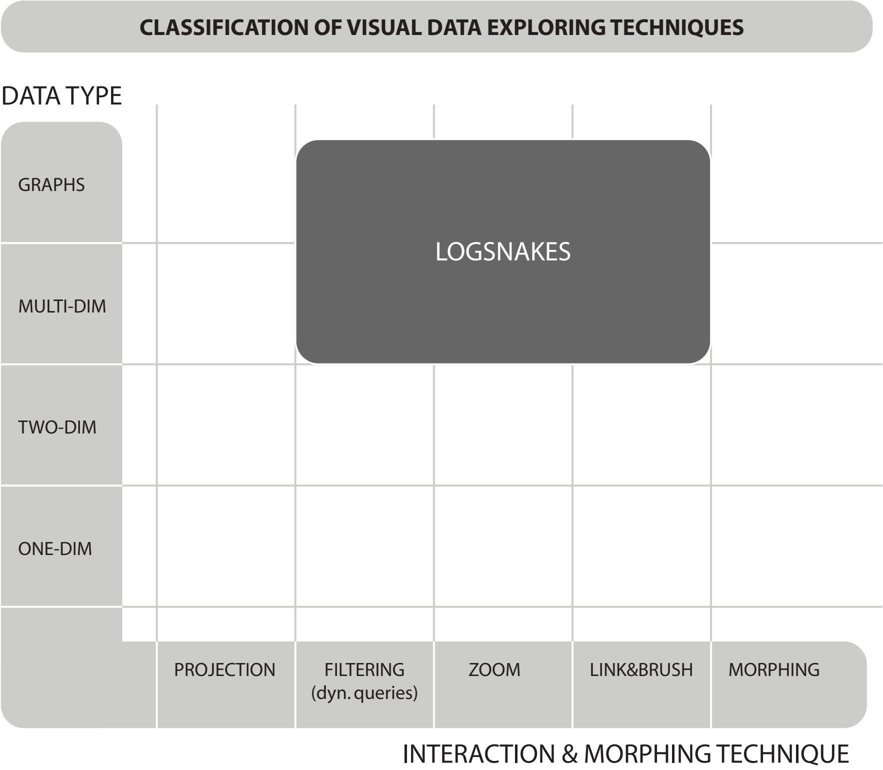


Figure 4.11: The possible features of the logsnakes method within the classification scheme.

To get an idea what the logsnakes are about, we present simulated data visualized with the logsnakes method in figure 4.12. From the figure, we can derive the following information:

- The length of the lag between positive attempts (does not necessarily have to be constant),
- the length of the lag between negative attempts (dependent on the strategy of the operator), and
- structure of users (maybe even groups).

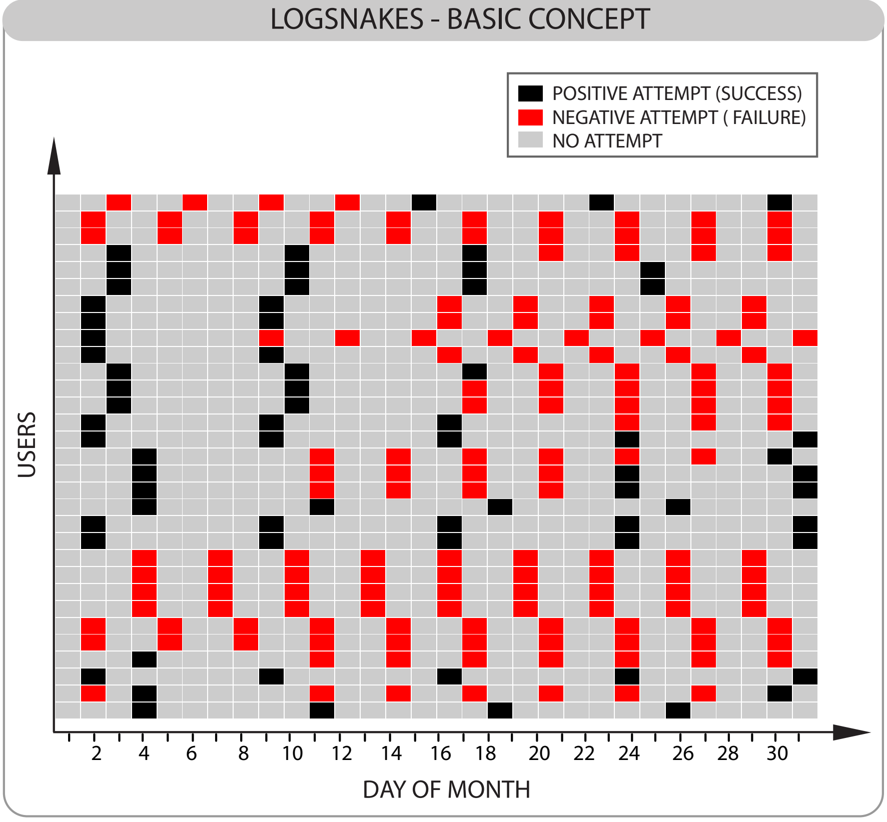


Figure 4.12: The plot shows the logsnakes of individual cases and their outcome. Moreover it gives an overview of the data.

The details we can find out of one *snake* is shown in figure 4.13.

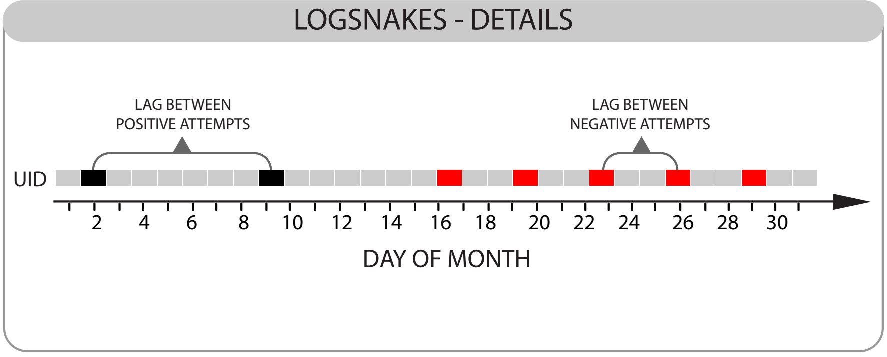


Figure 4.13: More information can be derived from logsnakes when a more detailed view on the results is applied.

4.3.2 Time Wheel

The time wheel is an idea which is derived from the SpiraClock from [Dragicevic and Huot2002]. Basically it tries to combine the possibility to show the aggregation level (which is a time interval) and its context. Therefore, it could also be considered to be a focus and context method (see [Aigner *et al.*2006] for details and further information).

The time wheel in the DICE implementation consists of the following elements (see figure 4.15 for a representation):

- **A calendar** which shows the current day, month and year. Within the calendar days can be marked with specific codes such as holidays or other relevant codes.
- **A clock** which shows the position of the playhead and the actual aggregation level.

The whole object is animated and moves forward or backward along with the playhead.

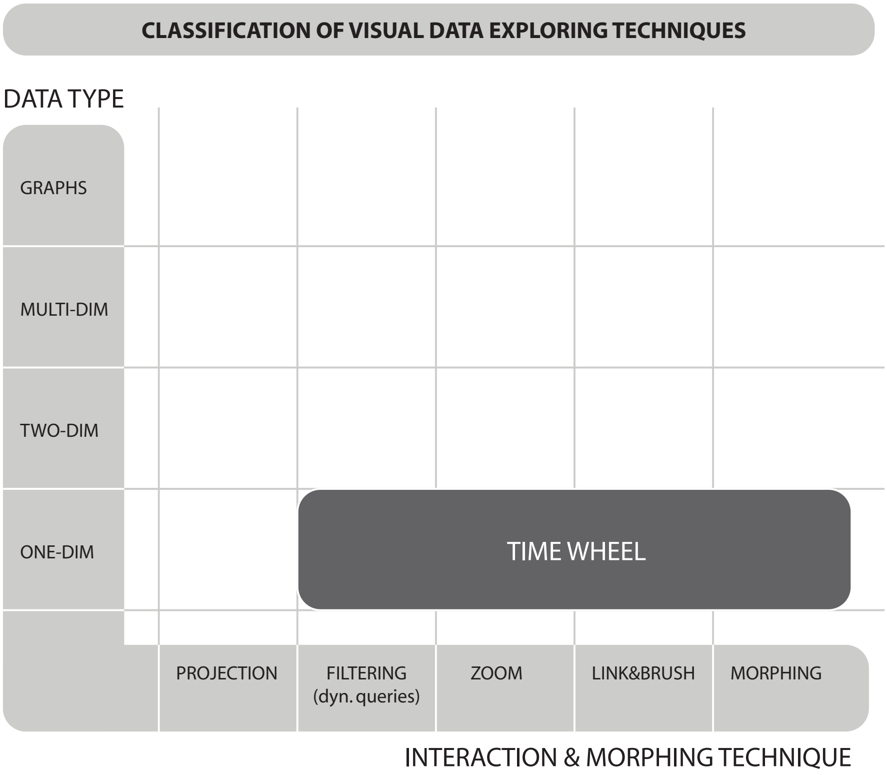
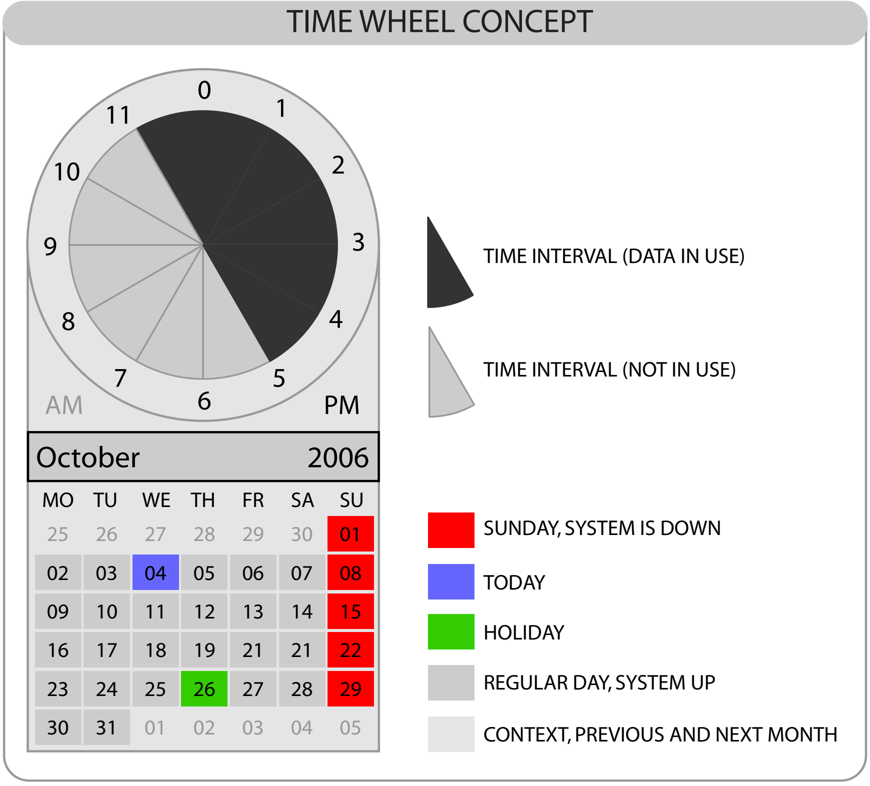


Figure 4.14: The possible features of the time wheel within the classification scheme.

The time wheel concept in figure 4.15 has the following components which we want to describe in detail:

- **The clock:** It shows the actual time interval (as selected aggregation level) which is used for the computation of the displayed figures as dark segments. Unused time intervals are displayed as light grey segments.
- **The Calendar:** The calendar represents the current day (according to the playhead of the timeline) in blue. Additional daily information is coded in other colors. One could

extend this principles to a more detailed resolution. This extension could allow events to start and end hourly and not only daily.



October 2006

MO	TU	WE	TH	FR	SA	SU
25	26	27	28	29	30	01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	21	21	22
23	24	25	26	27	28	29
30	31	01	02	03	04	05

Figure 4.15: The concept of the time wheel wants to show time related information in its context.

4.4 Conclusion

To summarize the chapter we want to point out that there is no such thing as the *perfect visualization method*. We want to propose statistical methods as enhancement of basic visualisation methods. This combination was presented as a fruitful one given the necessary statistical background of the method. Moreover a special method (logsnakes in section 4.3.1) was introduced, which was developed for a special task only. This should emphasis the necessity for the development of modular applications, which are able to easily integrate new and specialised visualisation methods.

Chapter 5

Implementation

> Before personalizing the products ... we should personalize the site in serving its users.
>
> - Myra Spiliopoulou -

5.1 Project Outline

5.1.1 Data Model

As stated in [Crad *et al.*1999] we need *data*, *a task* and *a schema* for knowledge crystallization (which is defined by [Crad *et al.*1999] as ,,getting insight into the data relative to the task”). We want to propose a datamodel which already pays respect to the tasks and to the schema. This results from the optimization of the datamodel for web based applications and the tasks within it.

The general data model for *active usage recording* for the DICE tool will look as follows:

variable	description
ID	The id of the record, an automatically incremented integer number.
User ID	An identical identification of the user. This must represent the user and serve as a (foreign) key for further user information.
Action ID	The action ID must represent the actions the user has performed. It must also serve as a (foreign) key to more information about the action.
Timestamp	A timestamp must make sure that the point of time, in which the action took place, is recorded.

Table 5.1: General variables for the data model which will suit active usage recording.

Success	This variable gives insight wether the (intended) action was successful or not. The basic case is a binary outcome, but it can be extended to more states (e.g., numeric codes).

Table 5.2: General variables for the data model which will suit active usage recording (continued).

User ID

The *User ID* represents the user, or the user profile, of the web application. If, for example, a family uses the same eBay account then one user id actually represents more than one person.

Action ID

This id represents the actions performed. Actions can be seen as *milestones* of processes.

Timestamp

The time, when an action was performed, is most important for the data model. It must be taken care that the time recorded is consistent.

Success

The actions performed can result into a state where they are considered successful (in terms of the web application and its processes). This variable can, like the action id, take several states (which might have different semantics). These semantics must have influence in the visualization, as they give deep insight in explanations of process results.

One can clearly see, that the basic data model contains a timestamp of some sort. We assume, that all data loaded into the DICE tool is *time oriented data*, for which certain conditions and possibilities apply.

We think the following definition for time oriented data is appropriate:

Definition - Time-oriented Data

„ *Data, where changes over time or temporal aspects play a central role or are of interest. [Aigner2004]* ”

Example - Data Model

The data model within the exemplary web application will look as follows:

- **User ID:** An incrementing integer, starting with 1.
- **Action ID:**
  - action: one in {RATE, VIEW, BUY}.
  - good: one in {ASSET, RINGTONE}.
- **Timestamp:**
  - h: The hour as integer, one in $\{0, \ldots, 23\}$.
  - m: The minute as integer, one in $\{0, \ldots, 59\}$.
  - s: The second as integer, one in $\{0, \ldots, 59\}$.
- **Success:** The (project) internal semantic of the success code. 1 represents *true success*, the other outcomes (one in $\{2, \ldots, 9\}$) represent different error codes.

One can easily see that the data model is not restricted to the first normal form of a relational database schema[1]. This is only true for the conceptual level. The data must of course be saved within a relational database schema which has all the requirements to suite up-to-date needs.

5.1.2 Identification of Actors

The DICE tool will be used for data investigation with visual and explorative methods. Thus there will be a defined set of persons (or roles) which are going to use DICE.

Analyst

The *analyst* is a person who is first and foremost interested in getting information and knowledge out of the data. This person is also very familiar with the web application and is interested in examining hypothesis. The person will certainly want to use an iterative approach, mainly following the *information seeking mantra* introduced by [Shneiderman1996].

This iterative approach defines the search for information as a ,,tour through the dataspace", in which patterns and coherence are hidden. This tour is defined by a variety of methods, which are connected and supplied by heavy interaction (see chapter 4). There opens a large space of combinations (which can be seen as *views on the problem*), which the analyst can apply.

The core use cases for the role of the analyst are:

- **Integrating the Data:** Which means to connect the data source to the DICE tool by defining the data interface as static source (e.g., XML file) or dynamic source (e.g., as database connection).
- **Analyzing the Data:** This is the essential task which makes use of all the possible methods and interaction techniques the tool provides. By doing so, the analyst can

[1] For more information on relational schemas one can look at [Kemper and Eickler2004].

approach different questions and generate, and maybe even proof, some hypothesis. These hypothesis should run in-sync with interesting and rewarding questions regarding the web application.

Example - Questions and Hypothesis

For the example given, possible hypothesis which should be answered by the tool could be:

- **H:** Profiled users are more active than unprofiled ones.
  - How many profiled and unprofiled users are there?
  - What is there absolute amount of actions?
  - What is the normalized amount of actions of profiled and unprofiled users?
- **H:** Midday is the most busy time of the day.
  - How are the actions distributed over the hours of the day?
  - What are the absolute counts of actions in this time?
- **H:** Advertisement helps to sell goods.
  - Which BUY actions were performed shortly after placing an advertisement?
  - Is there a significant difference?
- **H:** *X* is the best length for the lifetime of an advertisement.
  - How are the hits on advertisements distributed over the lifetime of the advertisement?
  - What are the absolute counts of actions in this time?

You will find these questions also in section about user feedback (2.3.5). To make it more practical, I want to show some case studies where one can see how a user of the DICE tool can possibly answer the questions from above.

(Method-) Developer

The role of a *method developer* will be crucial for the vitality of the tool. The DICE tool is connected to a vital data source. The underlying data model introduced in 5.1.1 provides a high quality data pool from web applications. Nevertheless there will be some or more details in the investigation cycles which might be unique.

To be aware that DICE meets such *dynamic demands*, it is the developers responsibility to use the tool with focus on new (or specially adapted) methods. For certain possible applications it might be necessary to apply a new technique. This task is maybe the most complicated one, because it does not only need a good understanding of the tool, but also a sense for the questions which might come up in the future. Changes in the raw data may also be of importance. It would be good if the developer would understand the data generating process to a maximum amount possible.

- **Method Development:** The main task of the developer will be the coding of new methods and the integration in the tool. There must be interfaces for sharing data with the other methods to provide interaction possibilities.
- **Feasibility Estimation:** To be able to identify and define methods and enhancements, it is important to estimate the potential realizability in terms of coding. I want to call this the *physical feasibility estimation*, because it reflects the actual implementation of theoretical proposals.

Maintainer

- **Feasibility Estimation:** To be able to identify and define methods and enhancements, it is important to estimate the potential realizability in terms of coding. I want to call this the *physical feasibility estimation*, because it reflects the actual implementation of theoretical proposals.

5.1.3 Multiple Views

Many graphical user interfaces are implemented as a window-like interface. All objects building a logical instance are represented by a distinct boundary. Within the DICE concept, major parts which are different as regards content are different views on the data. Those views are generated by using different visual techniques, which result into a graphical representation of the dataspace. In other words, those views are *single views* or more specific, *distinct views* when applying the definitions of [Baldonado *et al.*2000].

Definition - Single Views

,, *A* single view *of a conceptual entity is a set of data plus a specification how to display that data visually.[Baldonado* et al.*2000]* ”

Definition - Distinct Views

,, *We say that views are* distinct *when they allow the user to learn about different aspects of the conceptual entity. [Baldonado* et al.*2000]* ”

To be able to compare those graphics in an up-to-date manner, it is necessary to implement them in a flexible environment. This environment should also assist the *interaction* between the views in a beneficial way. The goal is to get information out of the data, and this goal should be aided by the environment. Given the definition of [Baldonado *et al.*2000], a multiple view system is the perfect match for the DICE concept.

Definition - Multiple View System

,, *A* multiple view system *uses two or more such distinct views to support the investigation of a given conceptual entity. [Baldonado* et al.*2000]* ”

The implementation of a multiple view system is very work intensive and can lead to a significant raise in costs [Baldonado *et al.*2000]. We argument now why the implementation of a multiple view system is important for the DICE tool. The argumentation is following the proposed rules *when to use multiple views* by [Baldonado *et al.*2000].

1. **Rule of Diversity:** *,,Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction or genres."* [Baldonado *et al.*2000]

    - The DICE tool has a data basis which allows diversity in many ways. The users all have identified user profiles, which allows user clustering.
    - There exist different levels of abstraction due to the aggregation level selected upfront.
    - Attributes can be identified through the Action ID from the datamodel (see 5.1.1).

2. **Rule of Complementarity:** *,,Use multiple views when different views bring out correlations and/or disparities."* [Baldonado *et al.*2000]

    - The mosaicplot (see section4.2.1) provides a graphical test for parameter independence.
    - Changes over time may cause a severe change in the resulting mosaicplot and thus change the outcome of dependencies.

3. **Rule of Decomposition:** *,,Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions."* [Baldonado *et al.*2000]

    - The timeline provides a tool to dynamically select different parts of the dataspace.
    - The aggregation level helps to identify structures on various levels.

4. **Rule of Parsimony:** *,,Use multiple views minimally."* [Baldonado *et al.*2000]

    - The user has full control over the amount of views presented.
    - Views are very distinct and thus manageable from the development point of view.

Figure 5.1 shows how a possible mockup of the DICE interface could look. The screen is divided in four main parts:

1. **Timeline:** As already said in section 4.1.3, the timeline is an integrated part of the tool and has its own area within the interface.

2. **View selection:** Buttons which represent the possible views and their status (on / off).

3. **Stage:** On the stage, the selected views are shown and arranged.

4. **Options:** The options panel presents options and settings for the selected view.

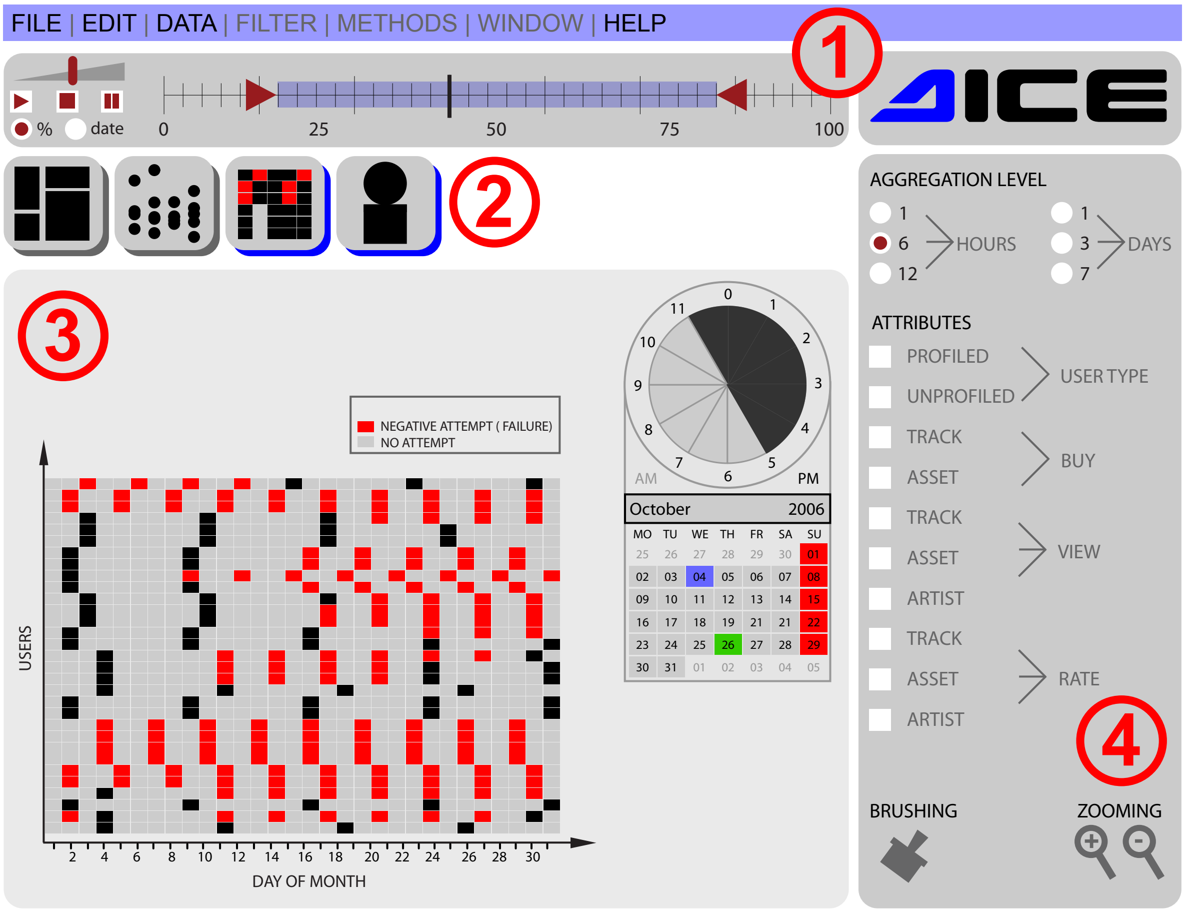


Figure 5.1: A mockup of the DICE screen, derived from the prototype.

5.2 Detailed Use Cases

Use cases are a highly valuable and widely used diagram type within the UML 2.0 specification that can help to model the possible cases of the usage of a software [Zuser *et al.*2004]. It can successfully show how people, which are involved in using the software, communicate with each other and the software.

The use case elements will be briefly described. The diagrams itself will be positioned in the appendix (section A).

There are three main connectors which are used to model the relation between use cases (from [Zuser *et al.*2004], see figure 5.2 for a representation):

- **Generalization:** This can be seen as the analogon to class hierarchies. The element at the shaft of the arrow is a specialisation of the element at the point of the arrow. The generalization is represented as an arrow with a *full black point*.

- **Extend:** Indicates, that the use case at the shaft can be extended by the use case at the point of the arrow. *Extend* and *include* connectors ar represented ar arrows with a light point and marked with a text specifying the type.

- **Include or Use:** Indicates, that the use case at the shaft does need the use case at the point of the arrow to be completed.

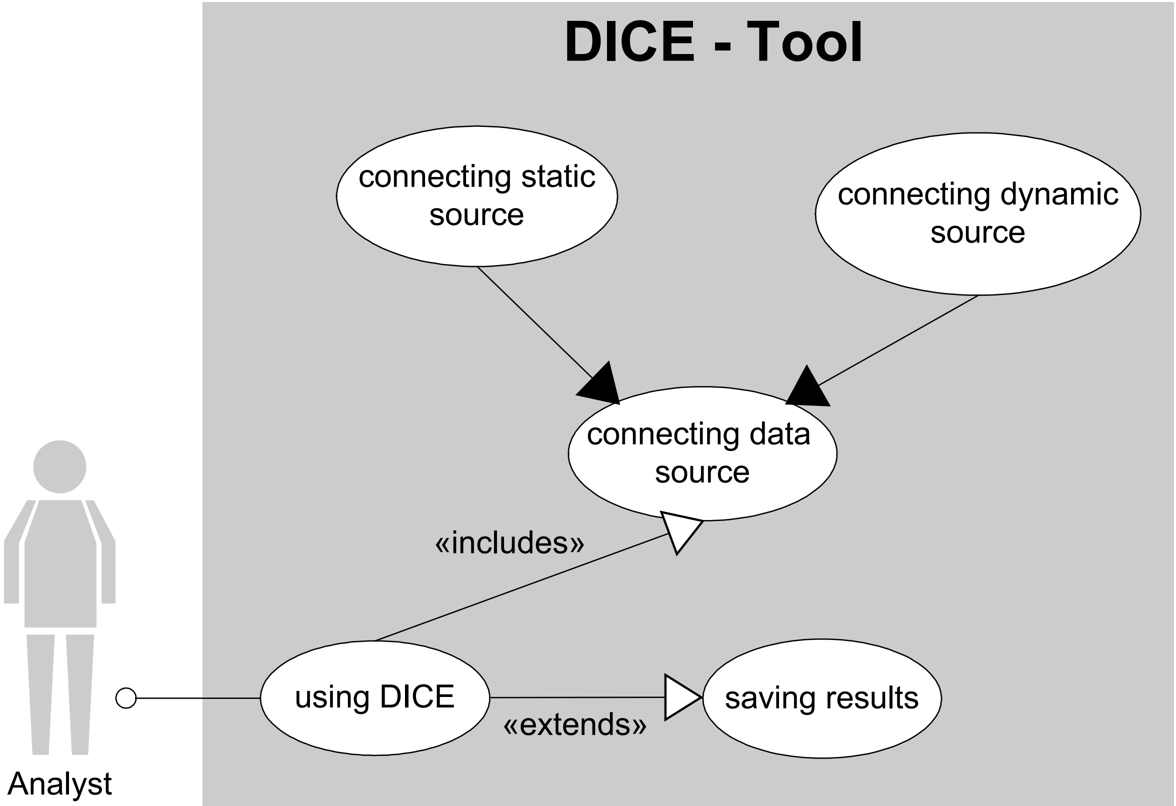


Figure 5.2: Use case elements from UML2.0 used to describe the DICE concept.

5.2.1 Analyst

Appendix A.1 shows the complete model of the uses cases for the analyst of the DICE tool. Starting from this ,,big picture", we want to give details to the most important use cases and describe them with a tablelike method (described by [Zuser *et al.*2004], pages 229 - 231). On this account we present a table in which we marked the use cases with their importance for the implementation (see tables 5.5, 5.6, 5.7 and 5.8). We then took the most important ones to further describe them.

use case	description	imp.
A: Data Analysis	The overall task of performing data analysis with the DICE tool.	**1**
A.1: Interact	The overall process of interaction. How we can use DICE to interact with the tool and its methods.	**1**
A.1.1: Dynamic Queries	The technique of using dynamic queries as described in chapter 4.	**1**
A.1.2: Zoom	Zooming as proposed in chapter 4.	**2**
A.1.3: Linking and Brushing	Linking and brushing as described in chapter 4.	**2**
A.1.4: Morphing	Morphing (e.g., of the mosaicplot) as described in chapter 4.	**2**
A.2: Aggregate the Dataspace	We need to aggregate the dataspace. Actions can only be investigated when we count their occurrence in certain intervals.	**1**
A.2.1: Timeline	The time line usage.	**2**
A.2.2: Aggregate via Time Interval	We must select a certain time interval for aggregation. Here we see how this selection affects other methods and use cases.	**2**

Table 5.5: The use case overview for the analyst giving details to task A.

use case	description	imp.
B: Prepare Data	Prepare the data for the usage within the tool. Most important step for providing a stable data source.	**1**
B.1: Connect to Datamodel	To secure the stable work of the database we have to make sure the variables provided by the data source (e.g., columns of the database) are successfully matched to the variables of the data model.	**2**
B.2: Connect Variables to Datamodel	Connecting the actual variables to the datamodel can make operations like splitting or aggregating source variables necessary.	**2**

Table 5.6: The use case overview for the analyst giving details to task B.

use case	description	imp.
C: Import Data	Import the data by connecting the tool to various data sources.	**1**
C.1: Choose and Configure Source	Choose between various data sources (e.g., SQL database, XML file) and configure the tool to ba bale to connect to it.	**2**

Table 5.7: The use case overview for the analyst giving details to task C.

use case	description	imp.
D: Visualize Data Space	Visualizing the data space means to use the methods (visualisation and interaction) and their combination to generate *views on the problem*.	**1**
D.1: Mosaicplot	The mosaicplot method described in section 4.2.1.	**1**
D.2: Wrapped Jitterplot	The wrapped dotplot with jittering for describing distributions in the data.	**2**
D.3: Logsnakes	Logsnakes are used to see how actions are related to the point of time they occur. The technique was specially developed for the DICE tool.	**1**
D.4: Time Wheel	The time wheel enables the analyst to observe the timely context of the actions and the views.	**2**

Table 5.8: The use case overview for the analyst giving details to task D.

A: Data Analysis

- **Primary Actor:**
  Analyst
- **Scope:**
  DICE Tool
- **Precondition:**
  none
- **Success Scenario:**
  The analyst has successfully prepared the data. He/she can interact with the proposed methods to generate different views on the data. The visualisation of the dataspace leads to significant insight in the problem.

  1. Data preparation as foregoing task.
  2. Interaction and visualisation are the core aspects for this use case.

- **Enhancements:**
  If the data could not be accessed, the task cannot be performed. The analyst has to make sure task B and C went successful.

- **Effects:**
  none

- **Technology:**
  The task makes use of the DICE tool. Core importance for the task has the multi window view of the tool.

- **Remarks:**
  Interaction and visualization in a multiple view application can slow down performance.

- **Open Questions:**
  none

B: Prepare Data

- **Primary Actor:**
  Analyst, Developer, Maintainer

- **Scope:**
  DICE Tool, Database, Data Repository

- **Precondition:**
  The variables, which can be derived from the data source match the data model to a sufficient amount.

- **Success Scenario:**
  All relevant variables of the data model can be connected to variables of the data source.

- **Enhancements:**
  none

- **Effects:**
  The tool would be provided with data.

- **Technology:**
  The task is performed within the DICE tool.

- **Remarks:**
  none

- **Open Questions:**
  none

C: Import Data

- **Primary Actor:**
  Analyst, Developer, Maintainer

- **Scope:**
  DICE Tool, Database, Data Repository
- **Precondition:**
  A database or data repository which has an implemented module for connecting the tool with the data source.
- **Success Scenario:**
  The data source can be successfully connected to the variables from the data model.
  1. Connect the data source to the data model by specifying the relevant variables.
- **Enhancements:**
  If the procedure fails, developers and maintainers have to make sure the routines for the connection of the data source are correct. Moreover it can be necessary to implement a special interface for the connection.
- **Effects:**
  Tasks A and D (and subtasks) can be performed.
- **Technology:**
  The usage of the DICE tool as well as the used programming language (e.g., JAVA) for implementing or adapting the interface.
- **Remarks:**
  The success of the DICE tool is very much dependent on how easy and intuitive it is to make use of vital data sources. The task of importing data plays a key role for the DICE tool.
- **Open Questions:**
  1. In which framework can a dynamic connection to a database be realised?
  2. Is a system, which selectively processes databases on runtime existing?
  3. How efficient would such a system be?

D: Visualize Data Space

- **Primary Actor:**
  Analyst
- **Scope:**
  DICE Tool
- **Precondition:**
  The task of data analysis has to be started, tasks A and B (import and prepare data) have to be completed successfully.
- **Success Scenario:**
  The success of this task depends on the aim of the analyst. Generally the scenario is successfully completed if the analyst finds new hypothesis or proof for existing hypothesis.

1. Generation of hypothesis.

2. Proof of existing hypothesis.

3. Basis for further analysis (subset of data).

- **Enhancements:**
  The task of visualization makes use of all visual techniques available. The connection between the visual techniques is provided by interaction techniques.

- **Effects:**
  The generation of hypothesis.

- **Technology:**
  The DICE modules.

- **Remarks:**
  none

- **Open Questions:**
  The visual techniques have to be implemented modular.

5.2.2 (Method-) Developer

Appendix A.2 shows the use cases for the method developer and the maintainer of the DICE tool. It tries to visualize the scope of the problem which lies partly inside the DICE tool (especially for the method developer) and the web application.

use case	**description**	**imp.**
A: Method Development	Development of additional methods (visual or interaction) for th DICE tool.	**1**
A.1: Suitable for DICE Interface	The method has to be suitable for the DICE tool in terms of the definitions of the interfaces.	**1**

Table 5.9: The use case overview for the method developer and maintainer presenting task A.

use case	**description**	**imp.**
B: Feasibility Estimation	Estimate if the proposed implementation is feasible with respect to the scope of the problem, the necessary variables and the compatibility to the DICE data model.	**1**
B.1: Scope of the Problem	Make sure the scope of the problem is suitable for the DICE environment.	**2**
B.2: Investigate Variables and Data Types	The DICE tool needs certain preconditions and has certain limitations regarding the data variables and data types.	**2**
B.3: Compatibility to the Data Model	Is the proposed method compatible with the data model of DICE? The data model is a basic precondition for every method within DICE.	**2**

Table 5.10: The use case overview for the method developer and maintainer presenting task B.

A: Method Development

- **Primary Actor:**
  Method Developer
- **Scope:**
  DICE API, DICE, Programming Language
- **Precondition:**
  none
- **Success Scenario:**
  The method was specified to a significant extent such that the DICE can handle input and output values and the method can be implemented as a DICE module. Success output would be:
  1. DICE method code.
  2. Library for the DICE tool.
  3. Tool update.
- **Enhancements:**
  The task must meet certain conditions from the DICE interface.
- **Effects:**
  The tool can be updated with another method.
- **Technology:**
  DICE source language
- **Remarks:**
  The module must not necessarily be written in the DICE programming language as long as the interface restrictions are conform.

- **Open Questions:**
  none

B: Feasibility Estimation

- **Primary Actor:**
  Method Developer, Maintainer

- **Scope:**
  DICE Tool, Web Application

- **Precondition:**
  none

- **Success Scenario:**
  The method proposed is considered to be feasible for implementation.

- **Enhancements:**
  The scope of the problem must be known and identified as well as the variable types and data types of the proposed method. Those details can help to gain information if the method can be implemented. One cornerstone of a successful development is the compatibility to the DICE data model.

- **Effects:**
  The implementation of the method can be granted.

- **Technology:**
  DICE Source Language

- **Remarks:**
  none

- **Open Questions:**
  none

5.3 Architectures

The following section tries to bring up possible architectures for the DICE approach. Prof. Hudec was very clear in his interview, that he thinks that there is no need to implement a whole new application. Due to this input we want to bring up alternatives for the realisation.

5.3.1 Server - Side

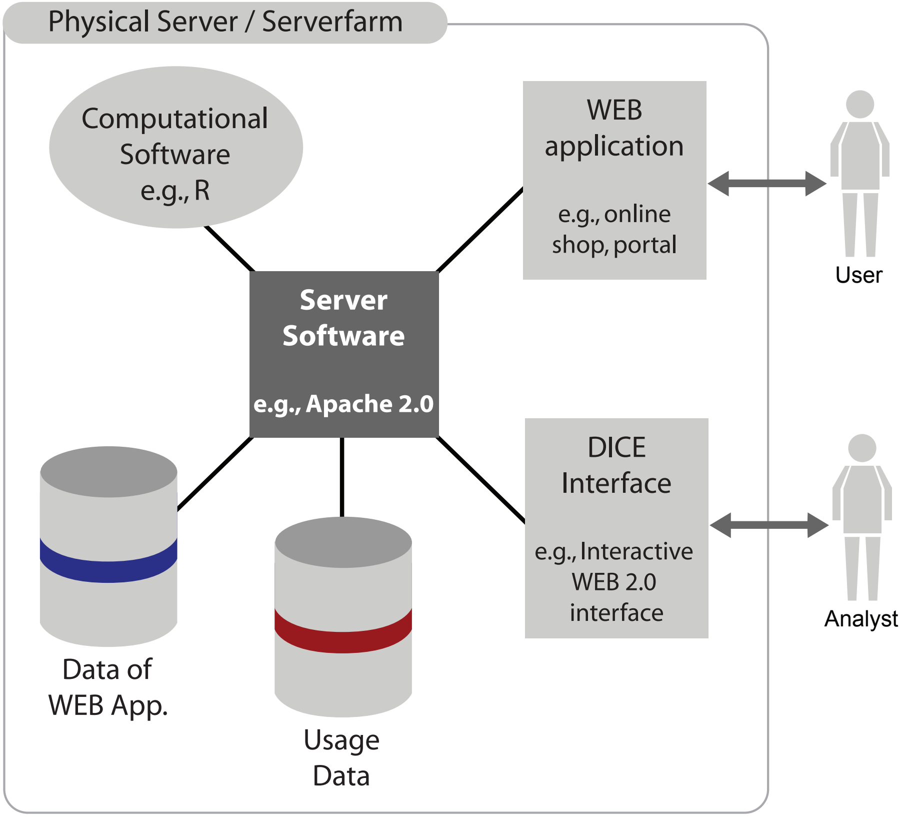

Physical Server / Serverfarm
Computational Software e.g., R
WEB application
e.g., online shop, portal
User
Server Software
e.g., Apache 2.0
DICE Interface
e.g., Interactive WEB 2.0 interface
Analyst
Data of WEB App.
Usage Data

Figure 5.3: A model of a server-side architecture for the DICE concept.

One possible architecture for the DICE approach could work as a server-sided software implementation. This would mean, that all the data could be stored at the server which serves the web application of interest. This would be mutual beneficial, because the data pool for serving the customers and writing the logfiles is centralized. On the other hand, the risk of default is higher since the data pools are not independent.

So we want to build a stable data pool which contains all the necessary data recorded by the web application itself. This is the typical application of a web server. It is not so clear if a server can also meet the up-to-date requirements of an interactive user interface as well as the computational power for the statistical methods.

part	description	implementation	scope
Server Software	The server side software for handling the services and requests.	Apache 2.0 Webserver	**WA**
Computational Software	The software which can handle the computational requests of the DICE interface. At the best, this software has an interface for connecting external applications (such as webserver software).	R	**DICE**
Data of Web application	The data from the web application is stored in a database.	mySQL, ORACLE DB, DB2, etc.	**WA**
Usage Data	The usage data of the web application investigated is also stored in a database.	mySQL, ORACLE DB, DB2, etc.	**DICE**
DICE interface	The DICE interface must meet certain conditions regarding the interactivity and dynamic power. This conditions restrain the suitability of a web based interface.	WEB 2.0 techniques (Java Script, HTML, CSS, PHP or Perl) as well as a Java Applet	**DICE**
Web Application	The web application which servers the users with services. These application is basis for the usage data.	WEB and WEB 2.0 techniques (Java Script, HTML, CSS, PHP or Perl)	**WA**

Table 5.11: Details for a possible web architecture of the DICE concept.

Let us now examine the elements of the server-side architecture in detail. First, we briefly describe the parts of the architecture, where a suitable solution is at hand.

- **Server Software:** Servers and their software are very sophisticated nowadays [2]. The interaction between web applications and databases is widespread and optimized to a great extent.
- **Web application development:** Web application exist in a very sophisticated way. Examples like Goolge Docs and Spreadsheets [3] or Goolge Mail [4] show, that web applications can be developed to act like local software. This benefit can be used to drive the implementation of the recording of high-quality usage data as well as tight evaluation cycles.
- **Web Databases:** Using databases in combination with a web server is a very common solution for almost every website or web application. Web applications, as defined in this work (see section 2.2), have to make use of a database.

[2] A list of most recent and most popular server software can be found at http://de.wikipedia.org/wiki/WebserverSoftware

[3] One can see http://docs.google.com/ for details.

[4] One can see http://mail.google.com/mail/ for details.

Now we examine the parts of the architecture, which are suboptimal to the proposed constellation.

- **Interactive Interface for Data Visualisation:** The interactive interface for the tool has certain conditions which must be matched. This criteria are complicated enough to fulfill even when the software is installed locally. As a local installation the software has more control about its surrounding characteristics (such as memory usage, disk space, etc.). If additional constraints, due to the implementation as a web based application, arise, it is even more complicated to meet the proposed necessities of DICE. Main issues would be:
  - Interactivity: Presenting figures which can be changed dynamically at runtime (as proposed by [Ahlberg *et al.*1992] with Dynamic Queries) is very hard (or not restrictable) in a web based surrounding.
  - Multiple Views: Multiple views are also quite hard to implement, because the setup of different views and the interaction between then is a tough challenge [Baldonado *et al.*2000]. Even more within a browser.
  - Dynamic Queries: To dynamically apply changes to all figures presented can be very cost intensive and challenging (see [Ahlberg *et al.*1992] for details).
- **Computational Software as Legacy Application:** Computational software is usually a standalone software package (cf. section 3.5) to be used on one PC only. Since the prices for this type of software are usually very high [5], there is no motivation to make this software accessible to a large number of users. Especially not by implementing it as a service of a web server[6]. But this is exactly the way the proposed architecture would need the software to act.

5.4 Conclusion

The chapter ,,Implementation" wants to show the cornerstones of the DICE tool. Starting from the important roles we presented overall use case diagrams and described the most important cases in detail. The resulting picture showed the areas which are vital for an implementation of the DICE tool.

The presentation and description of the use cases has shown that the interaction methods combined with several visualisation methods make the actors' tasks extremely complex. Thus it is necessary to provide a high quality data pool along with a stable data connection. If we make sure that these necessities exist, we can built a solid foundation for the developers and maintainers to perform the improvement and adoption of the DICE tools. Having secured a continuos evaluation would mean that the tool would constantly improve and gain flexibility.

Moreover, we presented architectures, in which the DICE concept could be implemented. The cornerstones were *a datapool*, *a computation engine* and *a flexible interface*. Putting all the issues together, one can find out the most appropriate way of developing DICE.

[5] For example, the Sawmill Enterprise Edition (for unlimited profiles) is 8.100 Euro (c.f. http://www.sawmill.net).

[6] Please note the difference here: the installation of a software as a service of a webserver is more than enabling the software to act via the internet. We are not talking about a software solution which has a web interface but about *featuring full application functionality via a web browser.*

Chapter 6

Evaluation

> Web-based applications ... require new approaches of evaluation.
> - Frank Heidmann -

6.1 Objectives

The objectives of this chapter are, to examine wether the use of the presented methods and their implementation in a tool such as DICE would be mutual beneficial. We want to show, that statisticians as well as people who use a lot of visual methods would be happy to find tools which combine the benefits of both worlds. Moreover we want to bring shortcomings of the proposed implementation and evaluate them on practical inputs from specialists whose daily work has to do with web usage analysis.

6.2 Methods

6.2.1 Interviews

For evaluation the thesis and the proposed work, we carried out several interviews with persons who are specialists on the field of concern. The interviews were recorded and the essentials are presented in the ongoing chapter. Details to the interview questions can be found in appendix B.1.

6.3 Participants

Professor Dr. Marcus Hudec

Professor Dr. Hudec [1] is part of the Department for Scientific Computing at the Faculty for Informatics at the University of Vienna. His main focus is on applied statistics in economic and social science. He is also member of the EC3 (electronic commerce and competence centre) in Vienna. At the moment he is coaching a project concerning web usage mining.

[1] More information can be found at http://homepage.univie.ac.at/marcus.hudec/ .

Jeffrey R. Horner BSc

Jeffrey Horner [2] is a system analyst at the Department of Biostatistics at the Vanderbilt University School of Medicine in Nashville, Tennessee. He presented one of his recent projects, the RApache project, at the Use!R 2006 in Vienna. The idea to try to connect R to a webserver and use it for the server-side architecture for DICE was self-evident. To be able to estimate the feasibility of this approach he kindly agreed on an interview.

Dipl.Ing. Erich Gstrein

Erich Gstrein is executive technician at the SAT research studios. He is responsible for the implementation of personalisation in web - and mobile - applications as well as the development of recommender systems. He can refer to a lot of experience in the field of the thesis, as he was a substantial part fo the development of the exemplary web application. His main focus was the individualisation of the platform with respect to user behaviour.

Florian Kleedorfer

Florian Kleedorfer is a student at the Vienna University of Technology and is working at the SAT research studios since 2005 as Junior Scientist. He also participated at the project concerning the online music platform described in [Gstrein2005] which is used in the work as exemplary web application. Within the project concerning the music platform Florian Kleedorfer worked on the feature extraction of possible distinction criteria for music files. This feature will be used for reasoning and decision support within the application.

6.4 Results

The results presented in the ongoing sections try to sum up the original versions of the interviews. You can find the extended versions of the interviews in appendix **??** and on the enclosed CD in folder `interviews`.

6.4.1 Reassurement

Web Information Cycle

The WIC was recognised as the state-of-the-art model for the path of information in the web. The circular approach of the WIC was confirmed as an enwidened model for the representation of the path of information with the inclusion of the feedback as integrated part.

Active Usage Recording and Data Quality

”Log files[3] are needed for the search for errors” was one of the statements of DI Erich Gstrein from the SAT studios. We talked about the necessity of the active recording of usage data. ”During the development (of the web application) we saw that the active recording of the users behaviour is necessary.” The SAT studios became aware of the importance of active usage recording when they saw how many things were missing, or defective, when evaluating the platform with their customer.

[2] Information on recent projects and the work of Jeffrey Horner at the Vanderbilt University School of Medicine can be found at http://biostat.mc.vanderbilt.edu/twiki/bin/view/Main/JeffreyHorner .

[3] Please note that the term log files refers to active usage recording (see section 2.3.5).

The conceptual development of a data model and the implementation of the scripts within the application payed off. Florian Kleedorfer assured us, that the data quality plays a fundamental role in the analysis of usage data.

The drawbacks of active usage recording (such as runtime delays or more efforts in data storage) are absolutely necessary for a up-to-date web application. "The customer demands supporting documents after the application is deployed." said DI Erich Gstrein. This documents should cover issues such as the benefit for customer and owner of the platform as well as performance. To keep the quality of those documents up and, at the same time, assure the ability of investigating usage behaviour with justifiable costs, active usage recording can strike a balance.

"For many cases (many web applications) the server sided (standard-) log-file is enough for basic evaluation." was a statement by Florian Kleedorfer, which emphasised that active usage recording is very much dependent on the complexity of the web application. DI Erich Gstrein endorsed this statement as well.

Prof. Dr. Hudec assured our statement, that *active usage recording* is a trend within more complex web applications. He brought up the main steps of collecting usage data as follows:

- Standard Server Log Files: Tracking the traffic and the server side usage.
- Pixel - Tracking: Using page impressions for tracking user actions.
- Active - Usage - Tracking: Writing the actions of users directly to a defined data model.

"The latter case will rise the data quality and lead to a better and more effective way in answering questions." assured Prof. Dr. Hudec the importance of this kind of data recording. Active usage recording is a clear trend in the field.

Prof. Dr. Hudec assured us, that the data model we propose can lead to a more formal way of data handling. Nevertheless he criticised the way the data model cuts out potential external user data (such as users behaviour outside the logon procedure).

Visual and Statistical Techniques

"The flexible approach of the DICE concept, by modular development for more or less specific tasks of data analysis, has a future. The alternative of having no application for visualisation is even worse." was another reassurement of the DICE concept from Florian Kleedorfer. In his opinion it pays to invest a little more to develop a framework which is flexible enough to allow external integration of additional methods.

The service of data visualization can be a service which is highly suitable for outsourcing. For DI Erich Gstrein there exist two valuable statistics for his context of use:

- Basic statistics which are provided by most web servers (see section 3.4.1) and
- How can relevant parameters for advanced statistics be found?

The second question ought to be answered by DICE. DI Erich Gstrein sees DICE as a development tool, which can be used for evaluating the site and "painting the broader picture of the dataspace". The DICE concept targets to present an overview on the data to gain ground for more detailed investigation of the data. Florian Kleedorfer emphasised the point that data analysis with specific methods are a lot of work, and therefor the necessity for providing pre-investigative visualisation is given. The DICE concept can easily answer simple hypothesis such as clarifying the potential influence of certain variables. Then the results can be quantified by more detailed techniques such as data mining techniques.

Prof. Dr. Hudec said, that the visualisation of results plays a very important role in the communication of the results of usage mining specialists. He brought examples where graphs of webpages were printed with click-probabilities for certain user groups. "Of course we use visual techniques there." he assured the question, wether visual techniques would be used to communicate results.

When asked about the ratio of presenting results compared to generating results (in data mining), Florian Kleedorfer answered: "The visual representation of the results of data mining processes is foremost for the presentation of the results to others." From his point of view the process of data mining would benefit a lot from the assistance of visual methods. Concluding the question, Florian said: "Not all algorithms are solving problems or substitute a good visualisation. If I need a clear graphical representation of results, it usually takes a lot of time."

Interaction Techniques

Answering the question of the practical relevance and benefit of interaction techniques, both, DI Erich Gstrein and Florian Kleedorfer, said that most presentations of data use very basic techniques. They found that this is due to the raising complexity of data visualisation and its explanation to nonspecialists. DI Erich Gstrein agreed that in the context he knows about, most of the time analysts perform tree like analysis. This means starting from some root, they step forward on demand, and start again from the upper node when the results are not as wanted.

This process is very time consuming and DI Erich Gstrein said: "I am thankful for every assistance to that problem!". The "workaround" of the iterative approach for information seeking is often performed *manually* by generating the figures one after each other. But as far as DI Erich Gstrein is concerned, there is almost no way of dynamically combine different views on the data within analysis tools. Pleas see figure 6.1 for a representation of the two concepts.

"Compared to the efforts of developing new views (on the data) manually, even a waiting time of several minutes would be acceptable." was Florian Kleedorfers opinion.

Prof. Dr. Hudec reassured the role of interaction techniques as a major field of research. The presented technique of dynamic queries is seen as important and vital for the analysis. Prof. Dr. Hudec mentioned the raising need for computational power with the raise of more complex data structures.

6.4.2 Shortcomings

Data Model

Prof. Dr. Hudec assured us, that the data model we proposed, is a minimum set of variables which allows *a more formal way of data processing.* The shortcomings is, that one reduces the amount of information.

One example Prof. Dr. Hudec brought up, was the use of restrictive data models within the evaluation of intranet web applications. There one has a restricted number of applications as well as a higher quality of data.

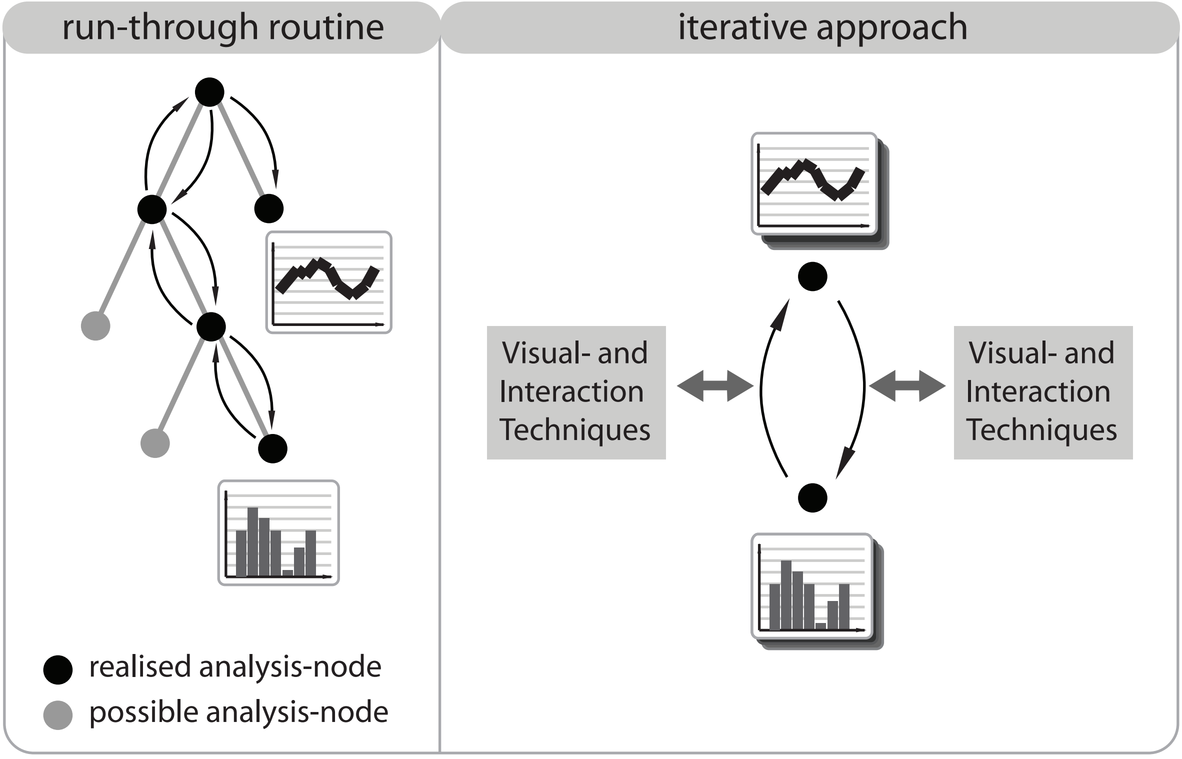


Figure 6.1: Two fundamental different approaches to data analysis.

Another example Prof. Dr. Hudec mentioned was his work with the Svarovski online shop[4]. "The shop has thousands of customers from all over the world every day, and around 4% identified customers." said Prof. Dr. Hudec. "If we cut off all the 96% customers to evaluate the shop, the whole procedure would collapse." This statement brought to the point, what we wanted to picture in section 5.1.1, when we mentioned that the datamodel must contain identification (key-) variables for a possible connection with other data sources. This issue is fairly major and must be taken into account.

Especially in the area of electronic tourism it is very important to be able to find a way to combine user sessions. There the actions performed would not necessarily be a whole process within the application. Thus the knowledge about the user behaviour can only be derived by sticking together the pieces, in this case, by product.

The mapping with larger amounts of data can be of great importance for the evaluation. Examples brought up by Prof. Dr. Hudec were

- the connection to the standard server log files,
- product databases of webshops,
- customer databases with additional data, and
- semantic information about site contents (which change over time)

[4] See http://www.swarovski.com for the implementation.

Flexible Structure

The modular approach of DICE is very beneficial for the application, from a theoretical point of view. There is a tradeoff between flexibility and feasibility, though. Florian Kleedorfer pointed out that the DICE concept must take care to be faster and easier than the potentially succeeding investigation (with data mining techniques), or the concept could lose ground.

Undoubtedly flexible software solutions are a good thing was Prof. Dr. Hudecs opinion. Nevertheless, he said that developing a new application would not be necessary. From his point of view a way of dealing with the DICE concept would cover these steps:

- define the task and the problems,
- define a sufficient data model, and
- use an existing data mining tool to evaluate the recorded data

In his opinion the existing data mining tools allow the customization and individualization of graphics and figures to a sufficient amount. This means, that by using standard graphics (such as histograms of graphs), the analyst can generate an overview of the data. To step into the dataspace more deeply, one can then use the provided interface and develop individual procedures to generate graphics.

Visual and Statistical Techniques

From DI Erich Gstreins's point of view the actual process of data mining does not need heavy graphical representations. He agrees with Florian Kleedorfer that the graphical representations are mostly to present *the results* of the data mining process.

This is a drawback for the DICE concept in some way. On the one hand the application of graphical representations throughout the whole investigation process could help even very sophisticated analysts. On the other hand, there is a point in saying that in such high level analysis, the effort is just not worth it.

Interaction Techniques

”If a system asserts a claim to perform at runtime, it can become annoying if it has hidden delays.” was Florian Kleedorfers clever remark on the topic of interaction techniques. We have brought up the issue in chapter 4.1 where we referred to [Hofmann2000] who was also investigating the *potential interactivity of a system* by responding times.

Use of Statistical Graphics

DI Erich Gstrein as well as Florian Kleedorfer see a high necessity in producing user-driven graphics. ”Graphics must be produced knowing the audience.” was the basic message here. Nevertheless, Florian Kleedorfer pointed out that for instance management persons have to become aware, that the data they use for decision making, is becoming more and more complex. So he thinks that also persons who do not necessarily have the ambition to understand more complex figures, have to think about it as a lack of advantage.

Prof. Dr. Hudec mentioned the need for the analysis of unstructured data (for instance texts, photos or videos in webpages). In his opinion this will develop more and more and can possibly become one of the major fields of research in the future.

6.5 Conclusion

The evaluation of the work was interesting and rewarding. Reassurements were mainly found in the area of the conceptual work, whereas implementations of DICE could be difficult. There are many ways to qualitatively implement software. Therefore, it is necessary to focus on the solutions which should be provided by the product.

Chapter 7

Conclusion and Future Work

> ” Having no future might be a kind of future, but without perspective. “
> - Wolfgang Kownatka -

7.1 Conclusion

The work can assist the goals identified in chapter 1 by giving a outline for possibilities within the field of visualization of data from dynamic web applications. It wants to focus clearly on easy-to-understand and easy-to-use methods, which enable developers and maintainers to receive hints for the enhancement of a web application.

What is named as *DICE concept*, is a framework which can be used for web usage mining with a strong focus on visualization and interaction. The concept is presented with a set of methods and techniques. In addition an exhaustive evaluation is done by interviewing experts as well as presenting state-of-the-art software.

7.2 Contributions

Contributions of a work are very hard to identify because the work has not achieved direct practical attention. Nevertheless, the work has been very close to a working example of many aspects presented (namely the exemplary web application). Thus we consider some parts of the work as helpful for theoretical as well as for practical issues.

In detail we want to point out three areas which are considered to be the most important areas of the work. Especially because they are of importance for personal work and studies.

7.2.1 Web Information Cycle

web information cycle—contributions The web information cycle is a complete and up-to-date representation of today's information flow in the web. During my studies and while browsing through several publications, I have not found a representation like it so far. Many existing representations were quite close to the one presented in this work, but still lack substantial parts.

We hope that the representation of the web information cycle can contribute to a better understanding of the information flows in modern web applications. Its aim is to combine flexibility with possible data input on the one hand, and stability through the representation as a *closed information flow* on the other.

The key aspect of the web information cycle, in our opinion, is the closed flow of information. All representations we found were presenting a one way picture. In our opinion this just does not take the cake. Modern web applications have a *cumulative repercussion*. This means that the processes performed in the application strengthen the users perception of the application. This fact should be modeled by closed flow of information of the web information cycle.

7.2.2 Active Usage Recording

Another important part of my work came up while we were was working with the exemplary web application. The *active recording of user actions* impressed me a lot. It can be integrated in almost any web application technology and can leads to an amazing data quality.

While investigating data from various sources, data quality has always been a major issue. At the data mining conference 2005 in Vienna, Prof. Dr. Hudec held a talk on how important data quality is for modern statistical analysis. By presenting a practical way of raising the data quality of web usage data, we want to provide a fundamental statement for modern web application developers. The time is here to use broadband internet connection also to serve the user via his trail, and not only in a direct way.

This can be identified as a trend. Modern web applications (e.g., Amazon.com) try to give hints to their users what to buy based on data actively recorded for that purpose. This is, because efforts for that type of service most likely result in higher payoffs. We want to point out, that also the web application itself can be the target of such active and direct exploration. The processes modeled within the web application are the *goods to sell* to the user. That this can help to achieve business related goals is trivial in our understanding.

7.2.3 Visual Methods

Visual methods are very interesting to develop and to explore. As a matter of fact, visual methods have a inverted relation between flexibility and adaptability. This relation is illustrated in figure 7.1.

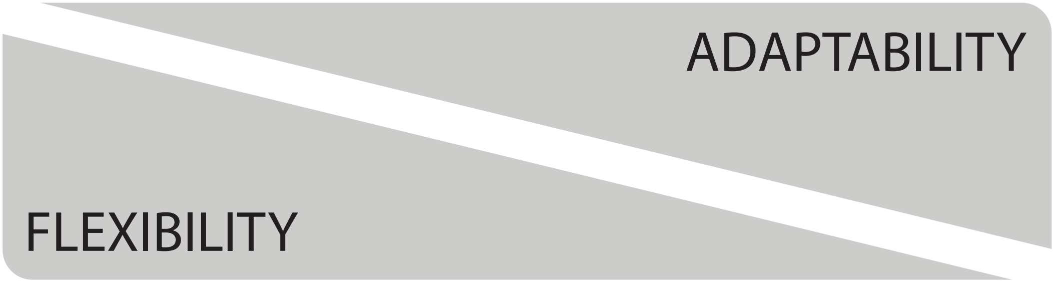


Figure 7.1: The inverted relation between felxibility and adaptability, among other things, makes the search for an optimal representation of the data non trivial.

As one consequence, we see the need to develop individual visual methods for special cases of exploration. This happened with the logsnakes technique within the development of this work.

On the other hand, there exist areas which can be used to derive interesting and rewarding ideas for visual methods. We tried to combine the field of visualization in some aspects to the field of explorative statistics. Especially with the mosaicplot we tried to give an example where fundamental statistics can be visualized in a great way.

Moreover we proposed to integrate the mosaicplot into several interaction techniques. In our opinion the integration of visual methods within flexible surroundings which allow interaction between methods should become state-or-the-art.

7.3 Ideas for Future Work

7.3.1 Active Usage Recording

The topic of active usage recording can be extended to set up several methods and functions for the easy implementation of active usage recording. Developing a framework for one major web development language (such as PHP) in which functions for active usage recording are available is possible. This would enable people, who develop web applications using PHP, to instantly use functions for active usage recording.

Moreover we extend this framework to other technologies where needed. This would help to integrate active usage recording in many web applications.

In Addition to functions for web technologies, modules for major open source web applications (such as content management systems) can be developed. The data will be of great use for developers of these applications.

7.3.2 Visual Methods

We try to constantly find visual methods with a statistical focus which can be used for information visualization. Most of the methods are very complex in terms of computation and calculations. Nevertheless, the identification and presentation of statistical methods within the field of visualization is a path to follow.

List of Tables

List of Figures

Index

Bibliography

[Ahlberg *et al.*1992] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM Press.

[Aigner *et al.*2006] Wolfgang Aigner, Klaus Hinum, Silvia Miksch, and Konrad Strutz. Infovis wiki - focus plus context, 2006. URL: http://www.infovis-wiki.net/index.php/Focus-plus-Context, Retrieved at: 29.10.2006.

[Aigner2004] Thomas Aigner. Visualization of time-oriented data. Vienna University of technology, Institute of Software Technology and Interactive Systems, Information Engineering Group, 2004.

[Ankerst2000] Mihael Ankerst. *Visual Data Mining.* dissertation.de - Verlag im Internet GmbH, Fritschestr. 68, 10585 Berlin, first edition, 2000.

[Arlitt and Williamson1997] Martin F. Arlitt and Carey L. Williamson. Internet web servers: workload characterization and performance implications. *IEEE/ACM Trans. Netw.*, 5(5):631–645, 1997.

[Baldonado *et al.*2000] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 110–119, New York, NY, USA, 2000. ACM Press.

[Baxley2002] Bob Baxley. *Making the Web Work.* New Riders Publishing, 1249 Eighth Street, Berkeley, CA 94710, USA, first edition, 2002.

[Bold and Neumeier2006] Max Bold and Franz Neumeier. Content power - zehn cms im vergleich. *INTERNET professionell*, 03/2006:50–53, 2006.

[Bolz2001] Christian Bolz. Web mining - software und dienstleistungen im vergleich. Universität Karlsruhe, Institut für Entscheidungstheorie und Unternehmensforschung, 2001.

[Boulos2003] Maged N Kamel Boulos. The use of interactive graphical maps for browsing medical/health internet information resources, 2003. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=149401, Retrieved at: 24.10.2006.

[Bultan *et al.*2005] Tevfik Bultan, Xiang Fu, and Jianwen Su. Tools for automated verification of web services. In *Automated Technology for Verification and Analysis: Second International*

Conference, ATVA 2004, Taipei, Taiwan, ROC, October 31-November 3, 2004. Proceedings, page 8. Springer Berlin / Heidelberg, 2005.

[Chen1999] Chomei Chen. *Information Visualisation and Virtual Environments.* Springer-Verlag London Limited, London, first edition, 1999.

[Christensen1990] Ronald Christensen. *Log-Linear Models and Logistic Regression.* Springer Verlag New York, 175th Avenue, New York, NY 10010, USA, second edition, 1990.

[Crad *et al.*1999] Stuart Crad, Jock Mackinlay, and Ben Shneiderman. *Reading in Information Visualization - Using Visions to Think.* Morgan Kaufmann, 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, first edition, 1999.

[DBMiner2006] DBMiner. Db miner, 2006. URL: http://www.dbminer.com/, Retrieved at: 16.10.2006.

[Dragicevic and Huot2002] Pierre Dragicevic and St&#233;phane Huot. Spiraclock: a continuous and non-intrusive display for upcoming events. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 604–605, New York, NY, USA, 2002. ACM Press.

[Emerson1998] John Emerson. Mosaic displays in s-plus: A general implementation and a case study. *Statistical Computing and Graphics Newsletter*, Vol. 9, No. 1, 1998.

[etracker GmbH2006] etracker GmbH. etracker - web controlling, 2006. URL: http://www.etracker.de/, Retrieved at: 13.10.2006.

[Fayyad *et al.*2002] Usama Fayyad, Georges G. Grinstein, and Andreas Wierse. *Information Visualisation in Data Mining and Knowledge Discovery.* Morgan Kaufmann Publishers, 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, first edition, 2002.

[Feng *et al.*2006] Guang Feng, Tie-Yan Liu, Ying Wang, Ying Bao, Zhiming Ma, Xu-Dong Zhang, and Wei-Ying Ma. Aggregaterank: bringing order to web sites. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, New York, NY, USA, 2006. ACM Press.

[Flesca *et al.*2005] Sergio Flesca, Sergio Greco, Andrea Tagarelli, and Ester Zumpano. Mining user preferences, page content and usage to personalize website navigation. *World Wide Web*, 8(3):317 – 345, September 2005.

[Flowerfire2006] Inc. Flowerfire. Sawmill, 2006. URL: http://www.sawmill.net/, Retrieved at: 18.10.2006.

[Friendly2001] Michael Friendly. A brief history of the mosaic display. Psychology Department, York University, Toronto, ON, M3J 1P3 Canada, 2001.

[Gstrein2005] Erich Gstrein. Online music portal. Smart Agent Technologies, Research Studios Austria, Vienna / Austria, 2005.

[Haigh and Megarity1998] Susan Haigh and Janette Megarity. Measuring web site usage: Log file analysis. *Network Notes*, 57, 1998. URL: http://www.collectionscanada.ca/9/1/p1-256-e.html, Retrieved at: 25.07.2006.

[Hao *et al.*2002] Ming C. Hao, Pankaj Garg, Umeshwar Dayal, Vijay Machiraju, and Daniel Cotting. Visualization of large web access data sets. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages 201–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[Hochheiser and Shneiderman2001] Harry Hochheiser and Ben Shneiderman. Interactive exploration of time series data. In K.P. Jantke and A. Shinohara, editors, *Discovery Science : 4th International Conference, DS 2001, Washington, DC, USA, November 25-28, 2001. Proceedings*, volume 2226/2001, pages 441–446. Springer Berlin / Heidelberg, 2001.

[Hofmann2000] Heike Hofmann. Exploring categorical data: interactive mosaic plots. *Metrika*, 51, Number 1:11 – 26, July 2000.

[Hong and Landay2001] Jason I. Hong and James A. Landay. Webquilt: a framework for capturing and visualizing the web experience. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 717–724, New York, NY, USA, 2001. ACM Press.

[Kappel *et al.*2004] Gerti Kappel, Birgit Proell, Siegfried Reich, and Werner Retschitzegger. *Web Engineering - Systematische Entwicklung von Web-Anwendungen.* dpunkt Verlag, Ringstrasse 19 B, D-69115 Heidelberg, first edition, 2004.

[Keim2001] Daniel A. Keim. Visual exploration of large data sets. *Commun. ACM*, 44(8):38–44, 2001.

[Kemper and Eickler2004] Alfons Kemper and Andre Eickler. *Datenbanksysteme - Eine Einführung.* Oldenbourg, Rosenheimer Strasse 145, D-81671 Muenchen, fith edition, 2004.

[Keogh *et al.*2002] Eamonn Keogh, Harry Hochheiser, and Ben Shneiderman. An augmented visual query mechanism for finding patterns in time series data. In T. Andreasen, A. Motro, H. Christiansen, and H.L. Larsen, editors, *Flexible Query Answering Systems: 5th International Conference, FQAS 2002. Copenhagen, Denmark, October 27-29, 2002. Proceedings*, volume 2522/2002, pages 240–250. Springer Berlin / Heidelberg, 2002.

[Kopka2000] Helmut Kopka. *LaTeX - Band 1: Einfuehrung.* Pearson Education GmbH Germany, Martin-Kollar-Straße 10-12, D-81829 Munich / Germany, third edition, 2000.

[Kosala and Blockeel2000] Raymond Kosala and Hendrik Blockeel. Web mining research: a survey. *SIGKDD Explor. Newsl.*, 2(1):1–15, 2000.

[Leung and Apperley1994] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, 1994.

[Liu *et al.*2000] Haifeng Liu, Wee-Keong Ng, and Ee-Peng Lim. Keeping a very large website up-to-date: Some feasibility results. In *Electronic Commerce and Web Technologies: First International Conference, EC-Web 2000, London, UK, September 2000. Proceedings*, page 399. Springer Berlin / Heidelberg, 2000.

[Markopopolous *et al.*2004] Panos Markopopolous, Berry Eggen, Amile Aarts, and James L. Crowley. *Ambient Intelligence.* Springer Verlag New York, 175th Avenue, New York, NY 10010, USA, first edition, 2004.

[Neumann2002] Horst A. Neumann. *Analyse und Entwurf von Software Systemen mit der UML.* Carl Hanser Verlag, Munich / Germany, second edition, 2002.

[NIST/SEMATECH2006] NIST/SEMATECH. e-handbook of statistical methods, 2006. URL: http://www.itl.nist.gov/div898/handbook/, Retrieved at: 21.10.2006.

[Notess2004] Greg R. Notess. The changing information cycle. *Information Today, Inc.*, Vol. 28 No. 5 Sep/Oct 2004, 2004. URL: http://www.infotoday.com/Online/sep04/OnTheNet.shtml, Retrieved at: 03.07.2006.

[O'Muircheartaigh and Payne1977] Colm A. O'Muircheartaigh and Clive Payne. *The Analysis of Survey Data - Vol2: Model Fitting.* John Wiley Sons Ltd., 111 River Street, Hoboken, NJ 07030-5774, USA, first edition, 1977.

[Raedt and Siebes2001] Luc De Raedt and Arno Siebes. *Principles of Data Mining and Knowledge Discovery.* Springer-Verlag Berlin - Heidelberg, Germany, first edition, 2001.

[Sala2006] Sala. Websites as graphs, 2006. URL: http://www.aharef.info/static/htmlgraph/, Retrieved at: 20.10.2006.

[Seo and Shneiderman2005] Jinwook Seo and Ben Shneiderman. Visualization key to external cognition in virtual information environments - a knowledge integration framework for information visualization. In *From Integrated Publication and Information Systems to Information and Knowledge Environments: Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday*, volume 3379 / 2005, pages 207–220. Springer Berlin / Heidelberg, 2005.

[Shneiderman1994] Ben Shneiderman. Dynamic queries for visual information seeking, 1994.

[Shneiderman1996] Ben Shneiderman. The eyes have it: a task by data type taxonomy for informationvisualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium*, volume Vol. 28 No. 5 Sep/Oct 2004, pages 336–343, 1996.

[Shneiderman2001] Ben Shneiderman. Inventing discovery tools: Combining information vizualization with data mining. In K.P. Jantke and A. Shinohara, editors, *Discovery Science : 4th International Conference, DS 2001, Washington, DC, USA, November 25-28, 2001. Proceedings*, volume 2226/2001, pages 17–28. Springer Berlin / Heidelberg, 2001.

[Software2006] ST Software. Ststat - real-time http reports and statistics, 2006. URL: http://users.hol.gr/ vtonic/ststat/frames.htm, Retrieved at: 16.10.2006.

[Spiliopoulou2000] Myra Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM*, 43(8):127–134, 2000.

[Srikant and Yang2001] Ramakrishnan Srikant and Yinghui Yang. Mining web logs to improve website organization. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 430–437, New York, NY, USA, 2001. ACM Press.

[Srivastava *et al.*2000] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, 2000.

[Tang and Shneiderman2001] Lida Tang and Ben Shneiderman. Dynamic aggregartion tu support pattern discovery: A case study with web logs. In K.P. Jantke and A. Shinohara, editors, *Discovery Science : 4th International Conference, DS 2001, Washington, DC, USA, November 25-28, 2001. Proceedings*, volume 2226/2001, pages 464–469. Springer Berlin / Heidelberg, 2001.

[Tergan and Keller2005] Sigmar-Olaf Tergan and Tanja Keller. *Knowledge and Information Visualization - Searching for Synergies*. Springer-Verlag Berlin - Heidelberg, Germany, first edition, 2005.

[Tiedtke *et al.*2002] T. Tiedtke, C. Märtin, and N. Gerth. Awusa - a tool for automated websites usability analysis. In *Design, Specification and Verification of Interactive Systems, Preproceedings of the 9th International Workshop DSV-IS 2002, University of Rostock/Universite catholique de Louvain*. Springer Berlin / Heidelberg, 2002.

[Wikipedia2006a] Wikipedia. Dynamic webpage, 2006. URL: http://en.wikipedia.org/wiki/Dynamic$_{W}eb_{p}age, Retrievedat : 23.07.2006.$

[Wikipedia2006b] Wikipedia. P-value, 2006. URL: http://en.wikipedia.org/wiki/P-value, Retrieved at: 29.10.2006.

[Wikipedia2006c] Wikipedia. Regelungstechnik, 2006. URL: http://de.wikipedia.org/wiki/DIN$_{1}9226, Retrievedat : 13.10.2006.$

[Wikipedia2006d] Wikipedia. Webapplikation, 2006. URL: http://de.wikipedia.org/wiki/Webapplikation, Retrieved at: 23.07.2006.

[Wikipedia2006e] Wikipedia. World wide web, 2006. URL: http://de.wikipedia.org/wiki/World$_{W}ide_{W}eb, Retrievedat : 20.10.2006.$

[Williamson and Shneiderman1992] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: evaluating dynamic queries in a real-estate information exploration system. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 338–346, New York, NY, USA, 1992. ACM Press.

[Wolf1999] Peter Wolf. Der jitterplot - konzept, 1999. URL: http://www.wiwi.uni-bielefeld.de/ naeve/software/revbook/revtmp/node98.html, Retrieved at: 28.10.2006.

[WUMproject2006] WUMproject. The web utilization miner wum, 2006. URL: http://hypknowsys.sourceforge.net/wiki/The$_{W}eb_{U}tilization_{M}iner_{W}UM, Retrievedat :$ 21.10.2006.

[Zuser *et al.*2004] Wolfgang Zuser, Thomas Grechenig, and Monika Koehle. *Software Engineering mit UML und dem Unified Process*. Pearson Education GmbH Germany, Martin-Kollar-Straße 10-12, D-81829 Munich / Germany, second edition, 2004.

[ZY Computing2006] Inc ZY Computing. 1-2-3 log analyzer, 2006. URL: http://www.123loganalyzer.com/google.htm?cpc-6, Retrieved at: 13.10.2006.

Appendix A

UML Diagrams

A.1 Use Cases - Analyst

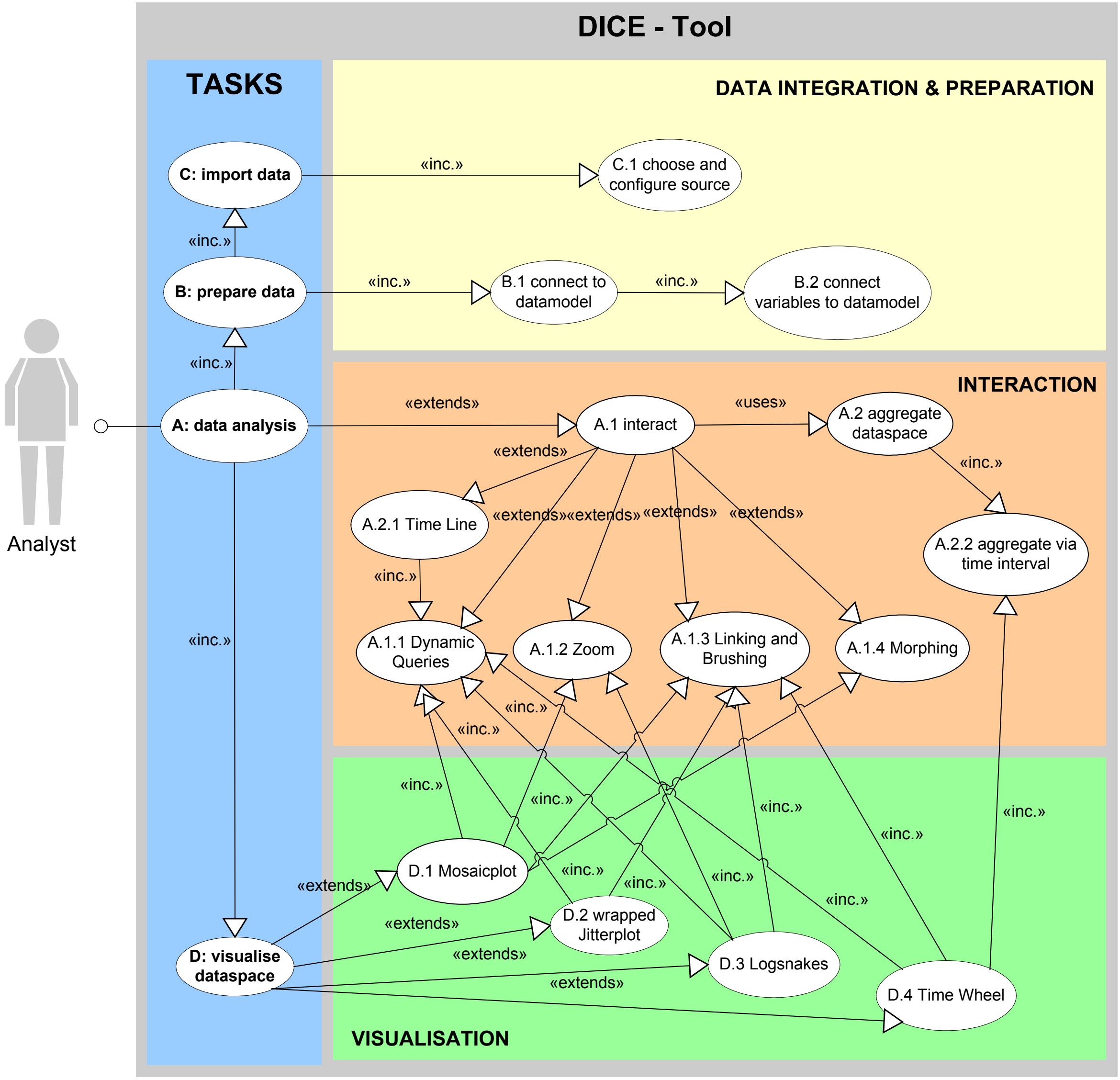


Figure A.1: use cases analyst - overview

A.2 Use Cases - Method Developer and Maintainer

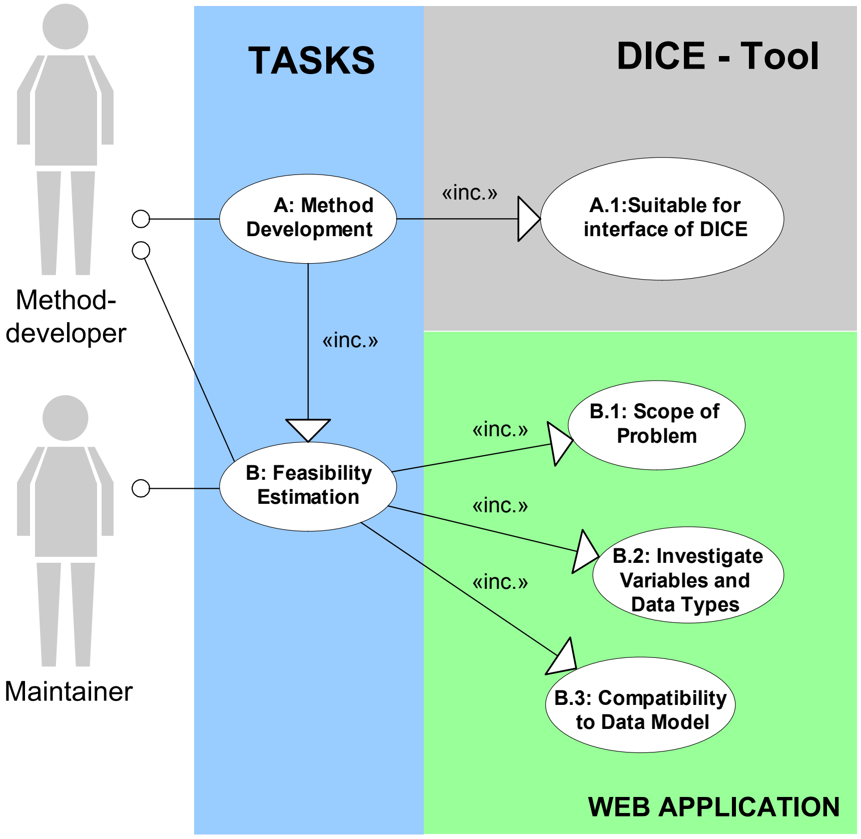

TASKS
DICE - Tool
A: Method Development
«inc.»
A.1:Suitable for interface of DICE
Method-developer
«inc.»
B.1: Scope of Problem
«inc.»
B: Feasibility Estimation
«inc.»
B.2: Investigate Variables and Data Types
«inc.»
Maintainer
B.3: Compatibility to Data Model
WEB APPLICATION

Figure A.2: use cases method developer and maintainer - overview

Appendix B

Interviews

B.1 Interview Guideline

B.1.1 Introductory Questions

- *Dear Mr. X, is it alright if I digitally record the interview?*
- *Is it also alright, if I place the recorded file on an enclossed CD to my thesis?*
- *Mr. X, would you please describe your related field of research?*
-

B.1.2 Web Information Cycle

- *Do you see a practical relevance of the Web Information Cycle?*
- *Could the WIC, in your opinion, be used for presenting evaluation iterations?*
- *I propose a trend towards (semi-) automatic feedback within web applications (WIC, part 4). Do you see this happening?*
-

B.1.3 Data Quality

- *With your experienced background in the field of web usage mining, how important is data quality for web usage mining?*
- *Could active usage recording have the potential to become a widely used practice in the area of web usage analysis? Or is it one already?*
- *Can you confirm, that active usage recording could avoind the following shortcomings (compared to standard server side data recording) such as: (see section 2.3.5)*
  - *The elimination of unreal users (due to caching).*
  - *Identification of erroneous requests.*
  - *Reduction in the amount of data recorded.*
-

B.1.4 Visual Techniques

- *What is your opinion towards flexible solutions, which allow (but also assume) the development of specialised techniques (such as Log Snakes)?*
- *Do you thinks visual techniques deserve more attention in the context of data mining and knowledge discovery?*
- *How many time (if possible in percent of an analysis procedure) do you spend to graphically represent your results (from data mining)?*
-

B.1.5 Interaction Techniques

- *Do you think the usage of interaction techniques for combining visual representations of data will gain importance in the future?*
- *I think of software solutions like GGobi*[1]*, which combine a lot of the proposed interaction techniques. Do you think those concepts will find their way into larger and more complex software solutions?*
- *Do you see that interaction techniques are applied in modern software solutions? Or is it more the algorithmic way of a ,,ru-through" routine?*
- *Can you confirm, that the iterative approach of seeking for information (overview first - zoom and filter - details on demand) has found its practical relevance (in data mining and knowledge discovery)?*
- *Talking about your experience, how long were those cycles for information seeking? I mean, do we talk about minutes to hours or more days to weeks?*
  Most certain the timespan varies regarding to the problem, but especially for web usage mining there are maybe limits.

B.1.6 Statistical Methods

- *From your point of view, do many statistical methods have the potential to be used for heavy interaction?*
- *From your experience, have you seen implementations making use of the interaction of statistical methods?*
- *Let us take the mosaic plot as an example. The mixture of underlying statistics and its graphical representation can be used in many visual contexts. Do you think examples like this can lead to a harmonisation between visualisation and statistics?*
- *Examples like the Parallel Coordinates introduced by A. Inselberg showed how explorative statistics and visualisation can be mutual beneficial. In your opinion, is the trend conservative or are the two diciplines getting closer?*

[1] For details please visit http://www.ggobi.org/

Appendix C

Usage Data

The presented Usage Data is randomly simulated data. There is *no* connection between the presented web application in [Gstrein2005] and the printed data. You can also find the presented data on the CD which, is coming alon with the thesis. Details for the CD content can be found in section ??.

```
     ID UserID ActionID_action ActionID_good Timestamp_h Timestamp_m Timestamp_s Success
1     1      1            RATE         ASSET           0          40          33       7
2     2      1            RATE      RINGTONE           0          30          18       7
3     3      1            VIEW         ASSET           0           6           5       1
4     4      1            VIEW      RINGTONE           0          23          42       8
5     5      1             BUY      RINGTONE           1           5          14       7
6     6      2             BUY      RINGTONE           1          33          34       3
7     7      2            VIEW         ASSET           1          29          45       7
8     8      2            RATE         ASSET           1          42          13       7
9     9      2             BUY      RINGTONE           1          26          35       1
10   10      2            VIEW         ASSET           2          42          31       6
11   11      2            RATE         ASSET           2          36          48       5
12   12      2            RATE      RINGTONE           2          56           1       8
13   13      3            RATE      RINGTONE           3          37          31       6
14   14      4            VIEW      RINGTONE           3          28           9       7
15   15      4             BUY         ASSET           3          11          27       4
16   16      4            VIEW      RINGTONE           3          37          38       5
17   17      5             BUY         ASSET           4          24          17       1
18   18      5             BUY      RINGTONE           4          20          13       8
19   19      5            VIEW         ASSET           5          25          55       6
20   20      5             BUY         ASSET           5          54          14       8
21   21      5             BUY      RINGTONE           5          51          33       7
22   22      6             BUY      RINGTONE           5          57          14       8
23   23      6             BUY         ASSET           5          15          14       7
24   24      6            VIEW         ASSET           5          26          24       6
25   25      6            RATE         ASSET           5          35           6       7
26   26      7             BUY      RINGTONE           6           5          34       6
27   27      8            VIEW      RINGTONE           6          10           8       8
28   28      8             BUY      RINGTONE           6          13          43       8
29   29      8             BUY      RINGTONE           6          17          53       1
30   30      9            RATE         ASSET           6          14          38       1
31   31      9            RATE      RINGTONE           6          15          33       3
32   32      9            RATE      RINGTONE           7          28          18       3
33   33     10            RATE      RINGTONE           7           3          56       6
```

34	34	10	BUY	RINGTONE	7	47	45	8
35	35	11	BUY	ASSET	7	55	17	2
36	36	11	BUY	RINGTONE	7	8	57	5
37	37	11	VIEW	RINGTONE	7	24	25	5
38	38	12	VIEW	RINGTONE	7	56	29	3
39	39	12	BUY	RINGTONE	8	20	35	4
40	40	12	VIEW	ASSET	8	7	56	2
41	41	12	BUY	ASSET	8	5	51	6
42	42	12	VIEW	ASSET	8	23	40	7
43	43	13	RATE	RINGTONE	8	30	55	1
44	44	13	RATE	ASSET	9	13	8	2
45	45	13	VIEW	RINGTONE	9	3	45	7
46	46	13	BUY	RINGTONE	9	9	51	6
47	47	13	RATE	RINGTONE	9	1	23	4
48	48	14	RATE	RINGTONE	9	46	24	1
49	49	14	VIEW	RINGTONE	9	25	32	3
50	50	14	BUY	ASSET	10	48	28	7
51	51	14	VIEW	ASSET	10	2	58	2
52	52	14	RATE	RINGTONE	10	0	39	5
53	53	14	RATE	ASSET	10	53	12	2
54	54	15	BUY	ASSET	11	46	41	8
55	55	15	VIEW	RINGTONE	11	49	9	8
56	56	15	BUY	ASSET	11	56	49	5
57	57	15	RATE	ASSET	12	21	19	7
58	58	15	BUY	RINGTONE	12	25	2	6
59	59	15	VIEW	ASSET	12	19	18	1
60	60	15	RATE	ASSET	12	35	51	4
61	61	16	RATE	RINGTONE	12	26	3	8
62	62	16	VIEW	RINGTONE	13	3	51	3
63	63	16	VIEW	RINGTONE	13	38	30	3
64	64	16	RATE	RINGTONE	13	15	52	6
65	65	17	BUY	ASSET	13	7	15	2
66	66	17	VIEW	RINGTONE	13	43	29	1
67	67	17	BUY	RINGTONE	14	27	2	4
68	68	17	BUY	RINGTONE	15	37	1	1
69	69	18	BUY	ASSET	15	6	27	7
70	70	18	RATE	ASSET	15	3	2	1
71	71	18	BUY	ASSET	16	5	56	2
72	72	18	BUY	ASSET	16	14	43	4
73	73	18	BUY	ASSET	16	22	16	5
74	74	18	VIEW	ASSET	17	12	4	6
75	75	19	RATE	ASSET	17	36	23	8
76	76	19	RATE	TRACK	17	13	37	5
77	77	19	VIEW	RINGTONE	17	45	13	2
78	78	19	VIEW	TRACK	18	21	24	4
79	79	19	VIEW	TRACK	18	31	51	4
80	80	19	RATE	RINGTONE	18	15	41	8
81	81	19	VIEW	RINGTONE	19	0	19	6
82	82	20	RATE	TRACK	19	19	40	2
83	83	20	VIEW	TRACK	19	10	5	1
84	84	20	RATE	RINGTONE	19	53	42	4
85	85	20	VIEW	TRACK	19	39	16	4
86	86	20	VIEW	RINGTONE	19	38	22	6
87	87	21	RATE	RINGTONE	19	26	13	7

88	88	21	RATE	RINGTONE	20	36	22	1
89	89	21	RATE	TRACK	20	4	6	1
90	90	21	RATE	RINGTONE	20	1	7	7
91	91	22	RATE	RINGTONE	20	43	40	4
92	92	22	RATE	TRACK	20	30	55	8
93	93	23	RATE	RINGTONE	21	21	48	5
94	94	23	RATE	RINGTONE	21	47	10	4
95	95	23	RATE	TRACK	21	54	46	5
96	96	23	VIEW	TRACK	21	42	19	3
97	97	24	RATE	TRACK	21	9	49	4
98	98	24	VIEW	RINGTONE	22	24	29	5
99	99	24	RATE	TRACK	22	38	34	4
100	100	24	RATE	RINGTONE	22	38	25	4

Appendix D

Code

D.1 Data Simulation

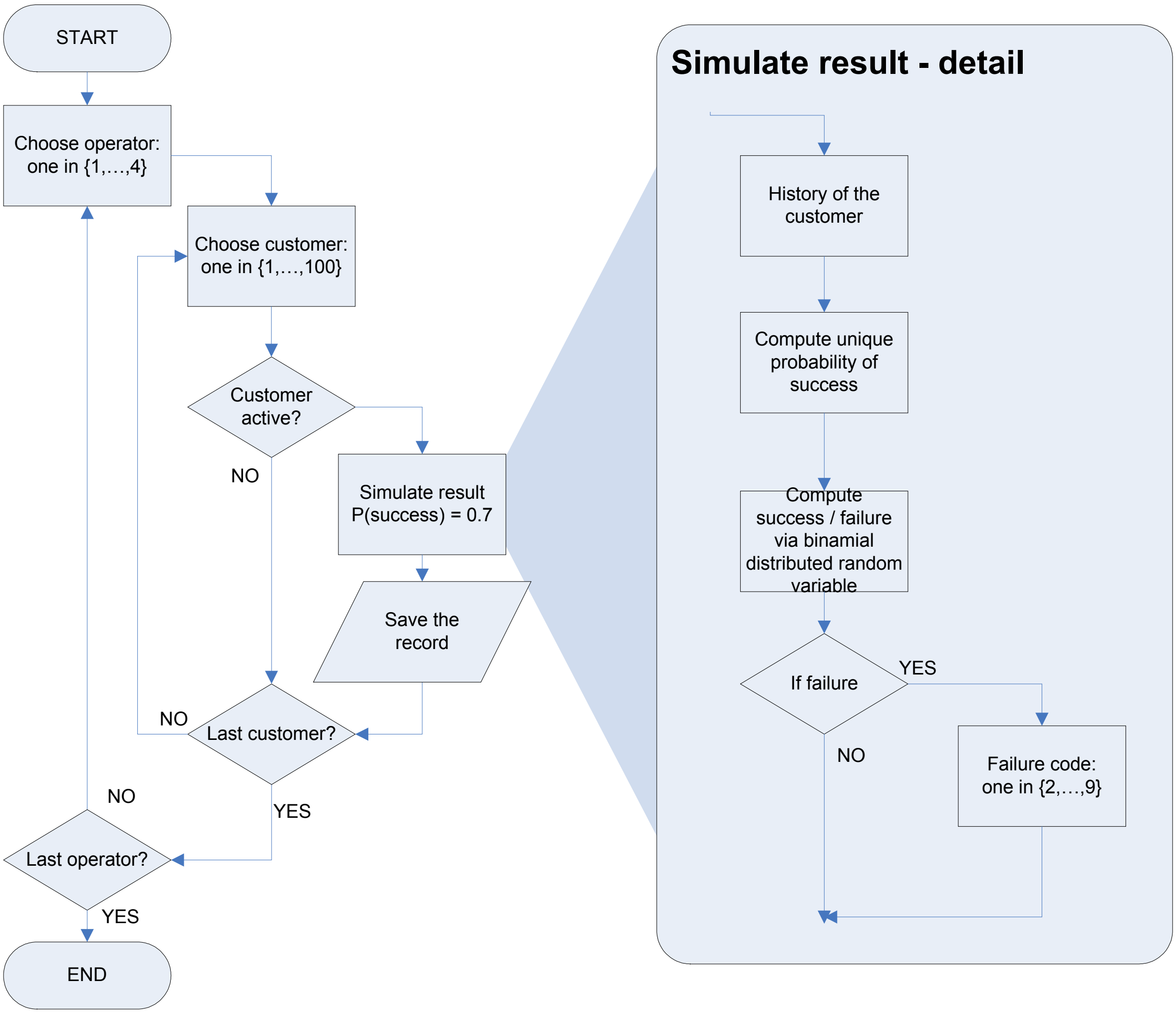

START
Choose operator: one in {1,…,4}
Choose customer: one in {1,…,100}
Customer active?
NO
Simulate result P(success) = 0.7
Save the record
NO
Last customer?
YES
NO
Last operator?
YES
END
Simulate result - detail
History of the customer
Compute unique probability of success
Compute success / failure via binamial distributed random variable
If failure
YES
NO
Failure code: one in {2,…,9}

Figure D.1: data simulation flowchart

```
users = data.frame(OID=c(rep(1,100),rep(2,100),rep(3,100),rep(4,100))
,UID=rep(c(1:100),4),STAT=rep(1,400))

# Datum als Dataframe zum Durchlaufen
# - d .. Tag: 1,2,...,365
# - T ... Tag
# - M ... Monat
# - J ... Jahr

dat = data.frame(
d=c(1:365),
T=c(c(1:31),c(1:28),c(1:31),c(1:30),c(1:31),
c(1:30),c(1:31),c(1:31),c(1:30),c(1:31),c(1:30),
c(1:31)),
M=c(rep(1,31),rep(2,28),rep(3,31),rep(4,30),rep(5,31),
rep(6,30),rep(7,31),rep(8,31),rep(9,30),rep(10,31),
rep(11,30),rep(12,31)),
J=rep(2006,365))

NEW_whichprob = function(U,O){
# calculates probability for a specific customer to
# have success on abo renewal (based on his/her history)
# UID .... User ID
# OPID ... Operator ID

if(exists(attempts)){ # check the history of the person
pres = attempts[attempts$UID==U&attempts$OID==O,] # results for the person
}else{
return(0.7)
}
}

whichprob = function(U,O){
# calculates probability for a specific customer to
# have success on abo renewal (based on his/her history)
# UID .... User ID
# OPID ... Operator ID
return(0.7)
}

getsuc = function(prob){ # invert the rbinom success value => 0 must be success
suc = rbinom(1,1,prob)
if(suc==1){
return(0)
}else{
return(1)
}
}

tryit = function(U, O){
# function models attempt to renew account
# U ... User ID
# O ... Operator ID
# return value:
```

```
# - 0 ... success
# - 1 - 8 ... failure (minor)
# - 9 ... user inactive!
if(users[users$OID==O&users$UID==U,3]!=0){ # user is active!
hisprob = whichprob(UID,OPID) # dependent prob. for success
suc = getsuc(hisprob)
if(suc==1){ # getting the detail of the bounceback:
suc = round(runif(1,1,9),digits=0)
}
if(suc==9){
users[users$OID==O&users$UID==U,3] = 0 # setting the user inactive!
}
}else{
suc=9 # send back inactivity
}
return(suc)
}

saveit = function(U,O,S,D){
# save the data of the attempt made
# - on the day D
# - for customer U (operator O)
# - with success S
# returns a vector with 8 values,
thisdat = dat[D,]
retval = c(U,O,D,thisdat[,2],thisdat[,3],thisdat[,4],111,S)
return(retval)
}

# TEST
saveit(3,3,0,34)
saveit(31,1,0,0)

# OPTIONS:
# waiting time to empty account:
wtmean = 20
wtsd = 5

for(i in c(1:4)){ # operators => O
for(j in c(1:100)){ # users => U
# user is set; now get the starting day;
day = abs(round(rnorm(1,mean=0,sd=wtsd),digits=0))+1
while(day<=365){
suc = tryit(j,i)
# save the results!
if(exists("attempts")){ # already saved some results!
attempts = rbind(attempts,saveit(j,i,suc,day))
}else{
res = saveit(j,i,suc,day)
attempts = data.frame(UID=res[1],OID=res[2],DAY=res[3],T=res[4],
          M=res[5],Y=res[6],ACT=res[7],S=res[8])
}
if(suc!=0){
# no success => try it again in 5 days!
if(suc==9){
day = 366
}else{
day = day+5
}
}else{
# success => try it again in random time
day = day+abs(round(rnorm(1,mean=wtmean,sd=wtsd),digits=0))
}
}
}
}

# NOW WE GOT THE DATA => VIZUALIZE!
dim(attempts)
\index{usage data simulation!R code|)}
\index{usage data simulation|)}
```

D.2 Logsnakes

```
attempts = read.csv("data.csv")

library(grDevices)
library(gregmisc)

logsnakes = function(dat){
# dat  ... data (attempts)
par(bg="gray")
plot(1,1,xlim=c(1,31),ylim=c(0,length(dat[,1])),type="n")
for(i in c(1:length(dat[,1]))){
# points(dat[i,5],i,pch=15,cex=0.5,col=(dat[i,8]+1))
# points(dat[i,5],i,pch="-",cex=0.5,col=(as.numeric(!dat[i,8])+1))
lines(x=c((dat[i,5]-0.5),(dat[i,5]+0.5)),y=c(i,i),col=(as.numeric(!dat[i,8])+1))
}
}

dat.o1 = attempts[attempts[,2]==OIDs[1],]
dat.o2 = attempts[attempts[,2]==OIDs[2],]
dat.o3 = attempts[attempts[,2]==OIDs[3],]
dat.o4 = attempts[attempts[,2]==OIDs[4],]
dat.o5 = attempts[attempts[,2]==OIDs[5],]
dat.o6 = attempts[attempts[,2]==OIDs[6],]

u5 = as.numeric(levels(as.factor(dat.o5[,1])))
u4 = as.numeric(levels(as.factor(dat.o4[,1])))


logsnakes(dat.o1[sample(1:length(dat.o1[,1]),size=10000),])

# just take the customers who have more attempts than one!

logsnakes.u = function(dat,users){
# dat  ... data (attempts)
par(bg="gray")
plot(1,1,xlim=c(1,31),ylim=c(0,length(users)),type="n")
i = 1
for(u in users){
u.dat = dat[dat[,1]==u,]
if(length(u.dat[,1])>1){
for(j in c(1:length(u.dat[,1]))){
lines(x=c((u.dat[j,5]-0.5),(u.dat[j,5]+0.5)),y=c(i,i),
          col=(as.numeric(!u.dat[j,8])+1))
}
}
#lines(x=c((dat[i,5]-0.5),(dat[i,5]+0.5)),y=c(i,i),
      col=(as.numeric(!dat[i,8])+1))
i = i+1
}
}

logsnakes.u(dat.o5,u5)
logsnakes.u(dat.o4[1:100,],u4)

\index{logsnakes!R code|)}
```