Master's Thesis

Towards Longitudinal Biomedical interactive Data Visualization on a Tablet Computer

Michael Schwarz, M.A. (Econ) BSc Bakk.rer.soc.oec.

Institute for Information Systems and Computer Media (IICM)
Graz University of Technology

Supervisor:
Assoc. Prof. Andreas Holzinger, PhD, MSc, MPh, BEng, CEng, DipEd, MBCS

Graz, March 2014

<This page intentionally left blank>

Masterarbeit

(Diese Arbeit ist in englischer Sprache verfasst)

Über interaktive, longitudinale Datenvisualisierung auf mobilen Tablets

Michael Schwarz, M.A.(Econ) BSc Bakk.rer.soc.oec.

Institut für Informationssysteme und Computer Medien (IICM)
Technische Universität Graz

Betreuer:
Univ.-Doz. Ing. Mag. Mag. Dr. Andreas Holzinger

Graz, März 2014

<This page intentionally left blank>

Abstract

Financial constraints limit the time per patient ratio in the medical sector. However, at the same time, a growing number of available medical treatment strategies should allow physicians to tailor therapies to patients' individual needs.

For such personalization of treatment, it is prerequisite to be well-informed about a patient's medical history. In most of the cases, this history is available as an array of longitudinal data points. To speed up the information perception process of medical staff, well-designed visualization of longitudinal patient histories plays an important role. Furthermore, this graphical information visualization supports the entire interaction process between physician and patient.

This thesis has two main objectives. First, it will provide an overview of methods of longitudinal data visualization and interaction on multi-touch tablet devices and, second, parts of these insights will be and incorporated into a practical section.

In this practical section, a report frontend of a clinical quality management system, which supports the decision-making process regarding the ideal therapy for patients with rheumatic disease, has been designed and developed for a mobile touch device (iPad).

Keywords (maximum 5)

Longitudinal data visualization, medical information systems, multi-touch interaction

ÖSTAT-Topics (maximum 4)

1138	30 %	1140	30 %	1157	20 %	1145	20 %

ACM Classification (maximum 5)

H.3, H.4, H.5, H.5.1, J.3

<This page intentionally left blank>

Zusammenfassung

Aufgrund der demographischen Entwicklung und der fortwährenden Etablierung von umfangreicheren Therapiemöglichkeiten ist unser Gesundheitssystem zunehmendem finanziellen Druck ausgesetzt. Indes zeichnet sich auch zusehends ein Trend zu einer individualisierten Medizin ab.

Eine wichtige Voraussetzung, die Ärzten eine derart personalisierte Gestaltung von Therapien erst ermöglicht, ist einen Überblick über die individuelle Krankengeschichte eines Patienten zu besitzen. In den meisten Fällen liegt diese Historie in Form von longitudinalen Datensätzen vor. Um den Informationsbeschaffungsprozess möglichst effizient und effektiv zu gestalten, ist eine gut gestaltete interaktive Datenvisualisierung sehr wichtig. Zusätzlich können diese Visualisierungshilfen herangezogen werden um den Interaktionsprozess zwischen Arzt und Patient dynamischer zu gestalten und damit die Servicequalität zu verbessern.

Diese Arbeit verfolgt zwei Hauptziele. Zum einen wird dem Leser ein Überblick über Methoden der interaktiven, longitudinalen Datenvisualisierung geboten und zum anderen wurden Teile des theoretischen Teils in eine praktische Arbeit übernommen.

Im Rahmen der praktischen Arbeit wurde das Report Frontend eines klinischen Qualitätsmanagement-System, das zur Unterstützung des Therapieprozesses von Rheumapatienten verwendet wird, als mobile Tablet-Applikation entwickelt (iPad).

Schlüsselwörter

ÖSTAT-Fachgebiete (Maximum 4)

1138	30 %	1140	30 %	1157	20 %	1145	20 %

ACM Klassifikation

H.4, H.5, H.5.1., J.3

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, 15th March 2010 ……………………………………………

First Name, Surname

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, 15. März 2010 ……………………………………………

Vorname, Zuname

Danksagung / Acknowledgments

Mein Dank gilt vor allem meinen Eltern, Geschwistern und Großeltern für ihre liebevolle Unterstützung während meines Studiums. Ihnen ist diese Arbeit gewidmet.
Bei meinen Betreuern, Prof. Andreas Holzinger und Dr. Klaus Simonic bedanke ich mich für die Betreuung und die hilfreichen Anregungen.

Abbreviations

HCI	Human–Computer Interaction
WIMP	Windows Icons Menus Pointers
GUI	Graphical User Interface
RCQM	Rheumatologic Clinical Quality Management System
ACF	Autocorrelation function
MA	Moving average
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
PACF	Partial Autocorrelation Function
UX	User Experience
UI	User Interface

Table of Contents

<This page intentionally left blank>

1. Introduction and Motivation for Research

1.1. Motivation

With today's growing number of possibilities regarding medical treatments, it becomes increasingly important for physicians and other specialized medical personnel, to find therapies, which are ideally tailored to the individual patient. However, a growing division of responsibilities in the medical sector makes it increasingly difficult for medical specialists to get a big-picture view on a patient's health status. Extensive and steadily growing electronic patient histories should – in theory – already be used to overcome this problem and support the task of decision making. In an ideal-case scenario, a physician should only decide on a therapy after a comprehensive study of these records. Nevertheless, financial constraints in the medical sector limit the amount of time physicians can spend on studying a health record for better decision making. It is therefore of uppermost importance, to make the information perception process of the decision maker with the aid of information technologies more efficient. Faster information perception leads to a better understanding of patients history, which ultimately enables an increase in therapeutic quality.

At the same time, the introduction of new devices with interactive surfaces allows users to interact with data visualizations in a more "natural" way (Lee et al., 2012). These devices allow the designer to replace overloaded menu-based interfaces with a more simplistic gesture-led interactive design. This transformation should benefit the information interaction process and thereby also improve the speed of knowledge discovery. Therefore, and also because of the mobile nature of these devices, which brings additional benefits, it is a logical step to include this technology in the everyday medical routine.

Most scientific evidence indicates that intelligent use of information technology enhances quality of healthcare. Numerous studies have been conducted to quantify this positive impact. In a literature review, Buntin et al. (2011) examined 152 studies on health care information technology. Their results show that 92 percent of these studies come to positive conclusions regarding the impact of information theory on health care.

According to a study of Amarasingham et al. (2009), greater automation of hospital information and the use of computer aided decision support systems can be associated with reduced rates of inpatient mortality, fewer complications, lower costs and shorter lengths of stay.

1.2. Objectives

This thesis has two main objectives. First, to give an overview of methods regarding longitudinal data visualization and interaction on touch-screen tablet devices and, second, to incorporate parts of these insights into a practical piece of work.

In the practical work, the report-frontend of a clinical quality management system (RCQM), which supports the decision making process regarding ideal therapy for patients with rheumatic disease, is developed for a mobile touch device (iPad). The primary aim of this development process was to speed up the information perception process by displaying the data in an organized and clear way, and allowing the user to quickly interact with the displayed longitudinal data by using intuitive multi-touch gestures. For interaction, we refer to the information seeking metaphor of Shneiderman (1994) as well as the interaction techniques listed by Schumann and Mueller (1999) and support gestures and functionality for overview, zoom-in, filter and zoom-out. Regarding visual presentation we follow the guidelines from Tufte et al. (1983) who define excellent statistical graphics as tools, which communicate complex ideas with clarity, precision and efficiency.

A further objective of the practical part was to increase the area of use for RCQM. By using mobile devices, new application scopes for RCQM are possible. For instance, the new mobile version enables physicians to quickly access patient data during home visits, display it and discuss further treatment together with the patient.

An additional aim was to build a relatively generic application, which can be used with minor additional effort to display other kinds of longitudinal data.

1.3. Structure of work

The thesis is structured as follows:

Chapter 2 presents the theoretical background for the practical work. The chapter is divided into three parts, representing different layers of abstraction.

First, longitudinal data visualization is discussed in detail. Afterwards, the differences between classical mouse-based desktop and mobile touch-device properties are highlighted. Finally, description and characteristics of medical information systems are introduced.

Chapter 3 presents related work to the later introduced practical implementation. This section is split into an introduction of RCQM, a clinical quality management system, which was developed for mobile multi-touch tablet devices, and in other related work-concerning medical information systems implemented on tablet devices.

Chapter 4 provides an overview of the implementation of RCQMmobile. This chapter briefly introduces the technologies used for implementation and gives an overview of some design decisions.

Chapter 5 summarizes the paper.

2. Theoretical Background

In general, the theoretical background that forms the foundation for the later introduced practical work can be divided into several categories. These categories can be seen as different layers of abstraction (Figure 1). In this chapter we will discuss all three layers.

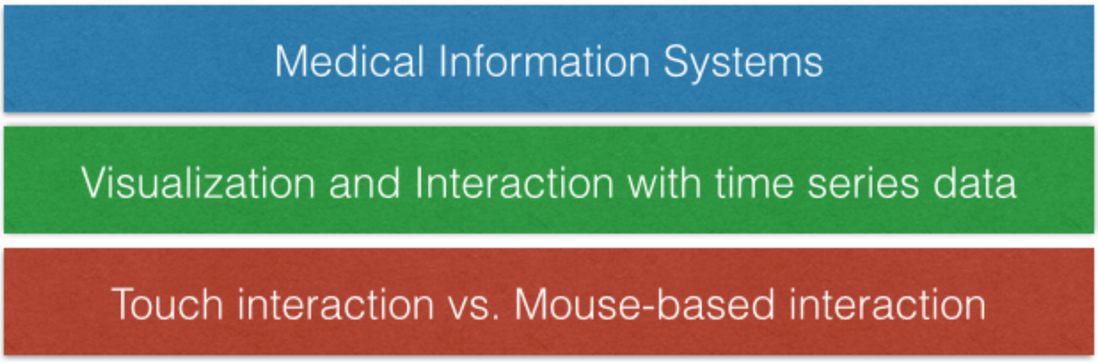


Figure 1 - Different layers of abstraction

On the highest level of abstraction the work is about medical information systems for handling longitudinal patient histories. Since the advent of mobile computing, there have been many approaches to bring health informatics on mobile devices. In this thesis we will provide a short presentation of some of these approaches and an overview of the definition of a medical information system

The second level is the scientific problem of interactive longitudinal data visualization and interaction. As the title of the thesis indicates, a large part of this chapter will deal with this topic. Thereby, not only visualization and interaction will be discussed, but also general properties of longitudinal data and time series modeling will be introduced.

Furthest down is the more low-level issue of device interaction properties. For this issue, we will investigate inter alia the question, which tasks can be enhanced with touch-input (compared to conventional mouse-based input).

2.1. Longitudinal data visualization

In more recent years, the growing ability to observe and record variables over time has led to a huge amount of raw longitudinal data. However, in most of the cases, this longitudinal data collection is just one step in a more substantial processing chain. More often, the ultimate reason for recording these arrays of data is to extract new knowledge by analyzing the recorded time series. For this analyzing step, time series visualization plays an important role.

Therefore, the core aspect of this chapter is to deliver an overview of different methods of time series visualization. But before doing so, it is necessary to make the reader familiar with some essential topics related to time series/longitudinal data. When working with time series, at least some of these topics will prove very helpful.

This chapter is divided as follows.

First, we will give an introduction into the definition and terminology of longitudinal data. This introduction will devolve into an overview of the mathematical methods for modeling time series.

Afterwards, sampling frequency will be discussed, which is an important topic, because a badly chosen sampling frequency can distort the whole time series analysis process.

We will then present the reason for and main objectives of time series analysis. Thereby, the topic of plotting will be introduced for the first time. Before presenting plotting in detail, we will also present the topic of time series transformation and dealing with missing values.

After the chapter on plotting, we will look into graphical methods for plotting value ranges and Tufte's definition of "chart junk".

At the end of the chapter, we will discuss the importance of longitudinal data within the healthcare sector and finally present some open-source visualization frameworks.

2.1.1. Definition and terminology of longitudinal data

Longitudinal data (or time series data) arises when a certain random variable is recorded as a sequence over time, whereas the measurement of some characteristics at (roughly) one single point in time is called cross-section data.

Depending on the measurement, differentiated distinction can be drawn between *discrete* and *continuous* time series.

A discrete time series is one in which the set T_0 of times at which observations are made is a discrete set.

On the other hand, if observations are recorded continuously over a time interval, e.g. $T_0=[0,1]$, a continuous time series arises (Brockwell/Davis, 2009). In contrast to analog recording, the process of digital recording is inevitably connected to a discrete sample frequency. Therefore, strictly speaking, although the lag between two measurements may be in the magnitude of milliseconds, we always have discrete time series data in the electronic data processing domain.

The same is true for the recorded value. To be exact, we always record discrete variable values. Nevertheless, the underlying nature of, for instance, height, blood pressure or weight of a person are examples for continuous variables, whereas the number of patient visits on one day is an example for discrete ones. For continuous values it is therefore important to choose a recording sampling frequency that draws an adequate picture of the underlying process.

Furthermore, for discrete time series, we can differentiate between *evenly* and *unevenly* spaced time series. Evenly spaced time series have constant time intervals between measurement points, whereas time intervals for unevenly spaced series can vary over time. Unevenly spaced time series are also called event-based time series (Warner, 1998), while evenly spaced series are called time-based records.

Most of the long-term clinical longitudinal data falls in the category of discrete, unevenly spaced longitudinal data, as time intervals between medical checkups vary in most of the cases.

An additional distinction regarding the recorded values can be made into *qualitative* and *quantitative* data.

Quantitative data is always numerical. It arises when certain characteristics are measured or counted. The number of patients in an ambulance is quantitative data, since it involves a count of the number of patients. Equally, the blood pressure of a patient is quantitative data, since the answer involves measuring the blood pressure.

Qualitative data is information that ranks or labels items, but does not measure or count them. For instance, if information about the drug name that is used for medication in a certain therapy is collected from patients, that data would be qualitative. If patients are asked during a medical checkup, whether they feel "very well", "'well", "average", "bad" or "very bad", their subjective health status is converted into a ranking. Therefore, qualitative data is generated also in this case.

Furthermore, depending on the recorded variable, measured values can be assigned to different type classes, namely *nominal*, *ordinal*, *interval* or *ratio* (Stephens, 1946). Thereby, the type defines the recorded data's level of structure. In general, qualitative data is either nominal or ordinal, whereas quantitative data is either interval or ratio data.

Nominal data is the type with the least structure. Its values are simple labels that cannot be ordered or ranked in a meaningful way. The name of the drugs given to several patients would be such kind of data.

In contrast, ordinal data can be ordered or ranked, but does not measure or count any data characteristics. Questions about, e.g. the subjective health status or satisfaction level, generally involve a ranking.

Interval data does measure or count any characteristics, but ratios between two measured values have no intrinsic meaning. This applies for measurement scales, where the zero point does not describe a state of absence of a quantity (e.g. the absolute lowest value on a scale). Temperature measuring in degrees of Fahrenheit is an example of interval data.

Ratio data means that ratios between two measurement points have an intrinsic meaning. For instance, if one patient has a dosage of 400 mg and another patient a drug dosage of 200 mg, the former has a dosage that is twice as high as the second one.

Longitudinal data can also be classified by the number of independent quantities that are recorded for each observation. If a physician examines a patient and only records the heart rate, the data has just one independent quantity and is called univariate. On the

other hand, data that involves more than one variable, is called multivariate. In special cases, when exactly two variables are measured, the data is called bivariate.
With regard to predictability of a time series we differentiate between *deterministic* and *stochastic* time series. If a time series can be predicted precisely, it is considered deterministic (e.g. if we look at the sinus wave). However, most of the time series fall in the category of stochastic time series. Thereby, future events are only partly determined by past behavior and exact predictions are impossible and must be replaced by the idea that future values have a probability distribution that is conditioned by knowledge of past values (Chatfield, 2003).
A time series is called *strictly stationary,* if the joint probability distribution does not change when shifted in time, i.e. $P_{t_1,\dots t_n}(x_1,\dots,x_n) = P_{t_{1+c},\dots t_{n+c}}(x_{1+c},\dots,x_{n+c})$.
A time series is called *weakly stationary* if $\mathbb{E}(x_t) = \mu$ and for the autocovariances

$$\mathbb{E}[(x_t - \mu)(x_{t-c} - \mu)] = \gamma_i \ , \qquad c = 0, \pm 1, \pm 2, \dots$$

This means, that the parameter mean and variance do not change over time, or follow any trends.

2.1.2. Modeling time series

Besides plotting time series, it is also important to be able to describe it in a compact mathematical form. This chapter will present some mathematical methods for modeling time series.
The goal of these models is twofold:
a) to explain the observed data and
b) to extrapolate the future development of the time series.
To be able to conduct any mathematical time series analysis, there needs to be some structure within the data. Without structure – such as overall trends or seasonal behavior – it does not make much sense to describe behavior of the data in a mathematical way. Nevertheless, it must also be remembered that, given the inherent stochastic nature of many natural phenomena, all models are an approximation of the true behavior of the system at best (Bisgaard/Kulahci, 2011). Therefore, it is assumed that data has a systemic pattern (which can be explained by our model) and a random noise (white noise) component, which hides the systemic pattern.

A basic model (regression on time)

As introduced in chapter 2.1.1, time series data can be interpreted as realization of a family of random variables $\{X_t, t \in T_0\}$. This suggests modeling the data as part of a realization of a stochastic process $\{X_t, t \in T\}\ where\ T\ \supseteq T_0$.

On of the simplest models is to assume that the time series varies around a constant value *b*. The variations are determined by a random variable ε_t.

$$X_t = b + \varepsilon_t$$

Thereby, the random variation ε_t, called noise, has a given variance and a zero mean. In this basic model, we assume that the noise variations in two different time periods are independent. These assumptions can be formulated as

$$E[\varepsilon_t] = 0, Var[\varepsilon_t] = \sigma_\varepsilon^2, E[\varepsilon_t \varepsilon_w] = 0\ for\ t \neq w$$

Including a linear trend into our assumption, we can enhance our first model into

$$X_t = b_0 + b_1 t + \varepsilon_t$$

These two models are special cases of a polynomial model.

$$X_t = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + \cdots + b_n t^n + \varepsilon_t$$

In all of these models, the considered time series is function of time and the model parameters.

$$X_t = f(b_0, b_1, b_2 \dots b_n, t) + \varepsilon_t$$

The value *f* is constant at any given time *t* and the expected value of ε_t is zero. Therefore we can conclude

$$E[X_t] = f(b_0, b_1, b_2 \dots b_n, t)\ and\ Var[X_t] = Var[\varepsilon_t] = \sigma_\varepsilon^2$$

This implies that there are two kinds of variability for a time series modeled in such a way. First, the mean value varies with time, and second, the distance from the mean varies stochastically. The only factor affecting the mean is time. All other random factors are subsumed in the noise component.

After selecting the model form, its parameters have to be estimated. The aim of this process is to find parameters that fit the available data points in the best possible way. Later on, these parameter values are used to make predictions about future observations. It is common practice to mark the estimated parameter values with hats.

$$\widehat{b_1}, \widehat{b_2}, \dots, \widehat{b_n}$$

The estimation also yields an estimate for the standard deviation of the noise, $\widehat{\sigma_\varepsilon}$. For the parameter estimation, most of the estimation methods put more weight on chronologically newer observations.

To illustrate this concept, we will apply a constant model to a given data set with 50 data points. Using all 50 data points, the estimate is given by

$$\hat{b} = \sum_{t=1}^{50} \frac{x_t}{50}$$

This estimation gives equal weight to all observations. If we want to give more weight to more recent data points, it is possible to include only these points to the estimation. Estimations using only the last *m* data points are called *moving averages.* To give some samples within the m-data point window more weight, multiplying or exponential factors can be applied. These methods are called "*Weighted moving average"* (WMA) and "*Exponential moving average"* (EMA).

This simple polynomial model is able to give a general overview of the trend of a time series. Nevertheless, it is problematic to assume that the only factor affecting the mean is time. A better approach is to assume that present realizations are functions of their own history. Such kinds of models are described in the following sections.

Modeling Autocorrelation

The assumption that present observations are functions of their own history is captured by the concept of autocorrelation. Autocorrelation is the similarity between realizations as a function of the time lag between them. Technically it is the cross-correlation of the time series data with itself. The models introduced in this section illustrate this behavior.

Generally, there are two main model classes for modeling time series processes with autocorrelation: moving average (MA) and autoregressive-models (AR). Based on these two classes, an array of mixed models (e.g. ARIMA, ARMA) exists.

The main difference between these two model classes is that MA models include lagged terms on the noise or residuals, whereas AR models include lagged terms of the time series itself.

One problem of time series modeling is to find the best model for a particular situation. Therefore, the first step in time series analysis should be to plot the observed data to get an impression of the overall process properties.

Box and Jenkins (1970) have introduced a model building process which helps to find the right model class. The process is illustrated in Figure 2.

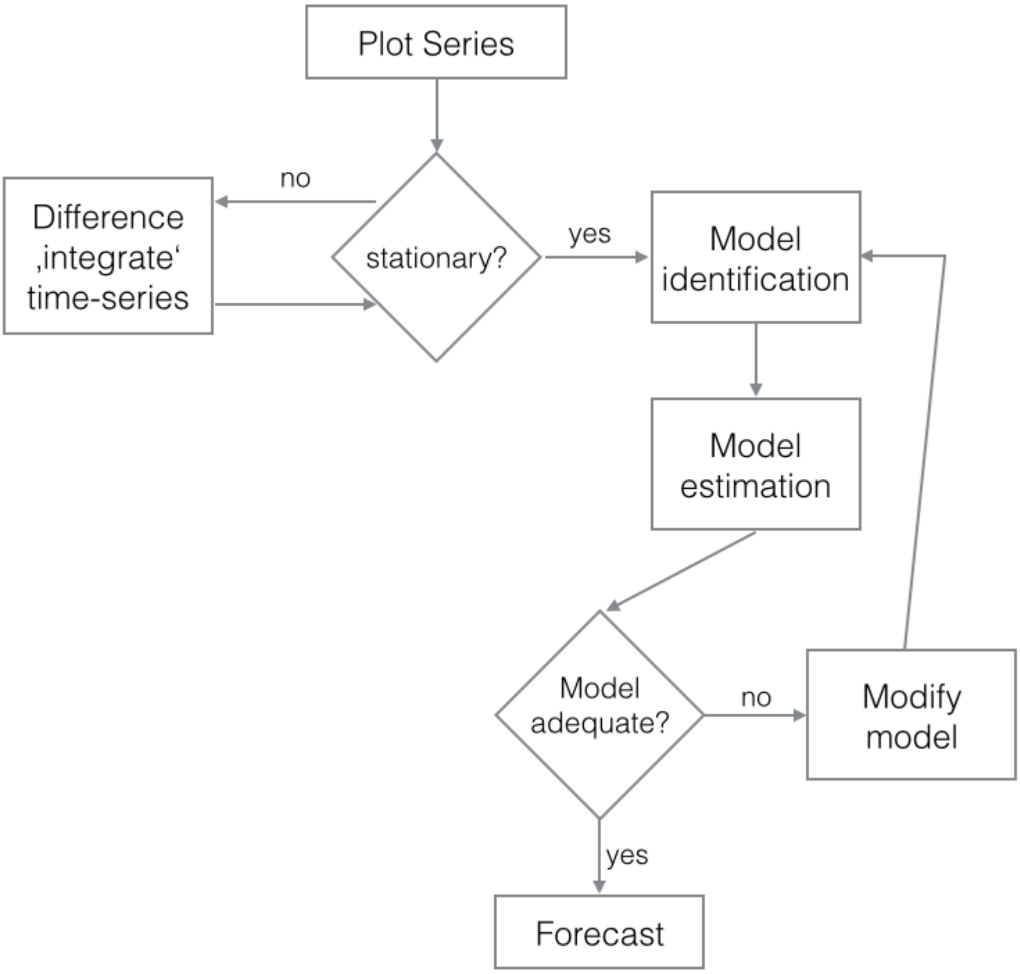


Figure 2 the Box-Jenkins model building process

As shown in the diagram, prerequisite for model identification is a *stationary* time series. However, for many processes, stationary can only be ensured by control actions taken at regular intervals or continuous maintenance of the system components (Bisgaard/Kulahci, 2011). Therefore, for many applications, non-stationary time series appear more interesting.

One solution to tackle this problem is to not focus on the original values that exhibit non-stationary behavior, but rather look at the changes between successive observations $X_t - X_{t-1}$. If these differences are stationary, we can go on by modeling the course of these differences. Otherwise, we have to take differences repeatedly, until we end up with a stationary time series.

For model *identification*, the autocorrelation function (ACF) and partial autocorrelation function (PACF) are crucial to the process.

The ACF is commonly used for finding repeating patterns within data. It is denoted by

$$\rho(k) = \frac{\gamma(k)}{\sqrt{\gamma(0)}\sqrt{\gamma(0)}}$$

in which $\gamma(.)$ is defined as the autocovariance function

$$\gamma(k) = E[(z_{t+k} - \mu)(z_t - \mu)]$$

The parameter k specifies the time lag for the calculation.

The PACF is the partial correlation function between the series and its lags over time. Given a time series z_t, the partial autocorrelation for the time lag k is the autocorrelation between z_t and z_{t+k}, without the linear dependence of z_{t+1} and z_{t+k-1}.

Plotting these two functions helps identify the right model. If the PACF displays a sharp cutoff while the ACF decays more slowly, an AR model should be used for modeling. Thereby, the PACF cutoff marks the number of AR terms (example shown in Figure 4). On the other hand, if the ACF displays a sharp cutoff and/or the lag 1 autocorrelation is negative, a moving average term should be considered. The lag at which the ACF cuts off is an indication for the precise number of MA terms (example shown in Figure 3).

Moving Average Model (MA)

Moving Average Models are one way to model univariate time series by incorporating information from past observations.

In this model, the time series is regarded as the moving average of a random shock series ε_t. A moving average model of the order "q" can be formulated as

$$X_t = \varepsilon_t + b_1\varepsilon_{t-1} + b_2\varepsilon_{t-2} + \cdots + b_n\varepsilon_{t-q}$$

Intuitively, the current value of a time series results from the weighted sum of past "q" errors plus a new error term ε_t. Usually, a moving average model is denoted as "MA(p)". For instance, a MA(1) model would look like

$$X_t = \varepsilon_t + b_1\varepsilon_{t-1}$$

and so forth.

Depending on the size of the parameter b, past shocks play a more or less important role for determining the present value.

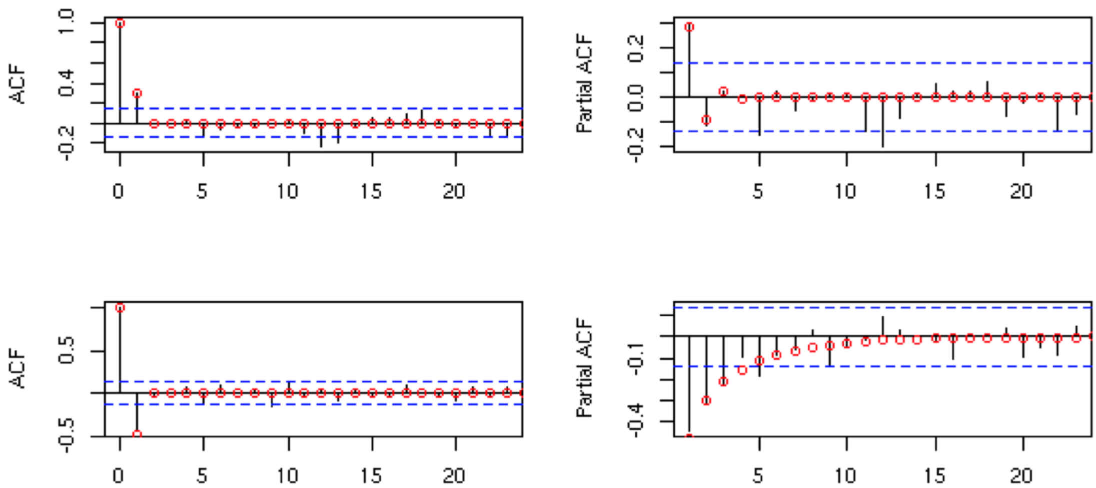


Figure 3 - ACF and PACF for a MA(1) process

(source: http://www.colorado.edu/geography/class_homepages/geog_4023_s11/Lecture16_TS3.pdf)

Autoregressive Models (AR)

In contrast to a plain moving average model, an autoregressive model specifies the current value by using linear combination of the preceding values and adding a stochastic error term. From a technical viewpoint it can be seen as a multiple regression model, where X_t is regressed on past values of X_t. Commonly, an autoregressive model of order "p2 is denoted as AR(p) and defined as

$$X_t = c + \sum_{i=1}^{p} b_i X_{t-i} + \varepsilon_t$$

Compared with the MA model, this model includes lagged terms of the time series instead of lagged error/noise terms.

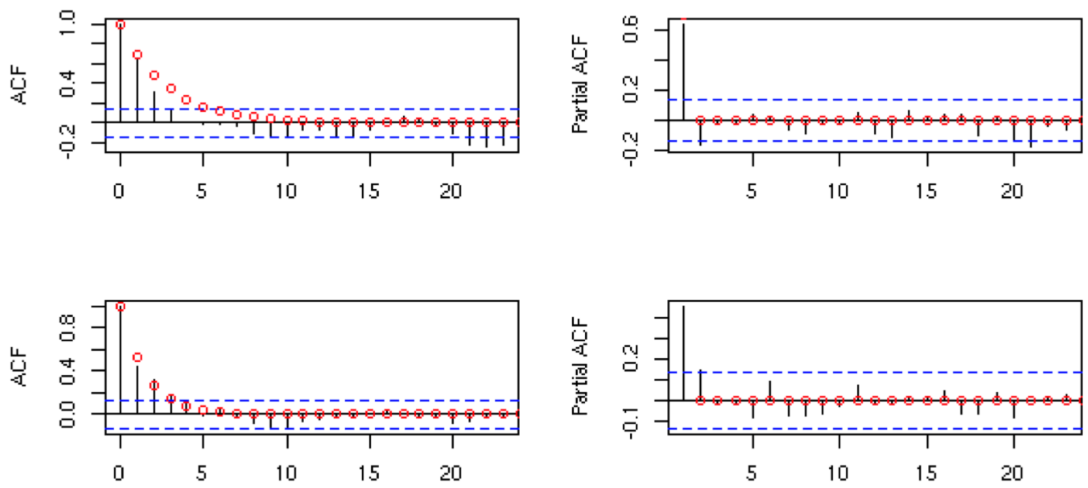


Figure 4 - ACF and PACF for an AR(1) process

source: http://www.colorado.edu/geography/class_homepages/geog_4023_s11/Lecture16_TS3.pdf

Autoregressive Moving Average Model (ARMA)

This model class includes a moving average as well as an autoregressive component. Using the notation of Box and Jenkins (1976), the model is summarized as ARMA(p,q). Thereby, the parameter p specifies the number of AR terms, while q stands for the number of MA terms.

For instance, ARMA(1,0) means that the model contains zero moving average parameter, and is therefore the same as AR(1). On the other hand, an ARMA(0,2) is the same as a MA(2) model. The model can be defined as

$$X_t = c + a_1 X_{t-1} + a_2 X_{t-2} + \cdots + a_{p1} X_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + b_2 \varepsilon_{t-2} + \cdots + b_n \varepsilon_{t-q}$$

Autoregressive Integrated Moving Average Model (ARIMA)

The ARIMA model is a generalization of the ARMA model, which incorporates the number of difference steps (for creating a stationary data set) into the model. Generally, the model is referred to as ARIMA(p,d,q), in which p stands for the number of AR terms, q for the number of MA terms and d for the number of difference steps.

Fitting the model

After choosing the parameters p, q and d, the model can be fitted by least squares regression. Thereby, the values are chosen in a way to minimize the error term.

Furthermore, it is best practice to choose the smallest values of p, q and d, that provide an acceptable fit to the data.

2.1.3. On the role of sampling frequency

Sampling frequency is the time interval between two successive observations (Warner, 1998). For instance, if patient values are recorded once a month, then the sampling frequency is $\Delta t = 1$ month. If blood pressure is recorded every 5 seconds, then $\Delta t = 5$ seconds.

One criterion for choosing the sampling frequency is the kind of patterns we expect to detect in the data. The shortest possible cycle, which can be detected in the data, corresponds to a cycle of two sampling distances. This can be derived, inter alia, from the Shannon-Nyquest Sampling Theorem (Jerri, 1977). This theorem says that if a function contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at series of points spaced 1/2B seconds apart.

The longest possible cycle can be up to N observations long, in which N is the length of the time series.

Another decision criterion regarding the sampling frequency is the lead-lag interval that might be important in multivariate time series. For instance, if a researcher thinks that a patient might respond to a special drug treatment within one day, the measurement should be sampled at least once a day (or even more often) to detect this relationship in the sampling data.

Choosing a false sampling frequency can lead to artifacts known as “aliasing”.

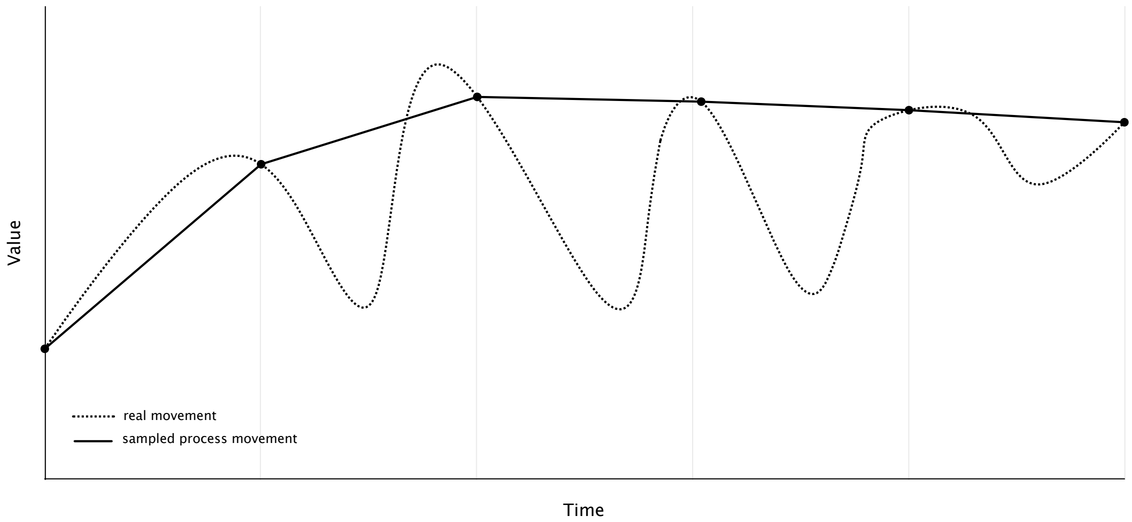


Figure 5 - Illustration of “aliasing”.

Figure 5 shows the problem in a graphical way. While the underlying process (real movement) shows fairly rapid cycles, the sampled line fully ignores this phenomenon. This happens because the sampling frequency is much slower than two times per cycle. The phenomenon is called “aliasing” because the lower frequency sampling masquerades, the higher the real frequency movement. The remedy for this problem is to choose a high enough sampling frequency.

This is more easily said than done. The ultimate problem is that it is not possible to detect aliasing by just looking at the recorded data. To determine, if aliasing is occurring, it is necessary to compare the recorded time series with the same time series recorded at a higher frequency. Only the higher-frequency recording would reveal the underlying shorter cycles. This means that in practice, the researcher must have knowledge on the frequency composition of the underlying process that is being sampled. Furthermore, the researcher has to assess, which granularity of movement is of interest for his research.

If doubt in respect to the sample frequency persists, Warner (1998) suggests to sample at the highest feasible frequency, in order to allow an evaluation, if higher frequencies exist. If unwanted high frequencies are present, these frequencies can be filtered out by averaging together or smoothing the observations to get a time series with fewer observations.

Furthermore, it must be mentioned that in case of event-based time series the statements regarding frequency need to be interpreted in terms of events instead of time units.

2.1.1. Objectives of Time Series Analysis

Study the past, if you would divine the future

- Confucius (551-479 BC)

As already said by Confucius roughly 2500 years ago, a main reason for studying the past is to divine something about the future. What certainly sets us apart from Confucius is that nowadays we are able to extend our analysis of past data points with computational power into new scale.

But is time series analysis a meaningful tool for making inferences on the future behavior of a process? After all, past development is studied in a very partial form.

Seen from a strictly economic point of view, it clearly makes sense. Economically, the only permissible reason for allocating current valuable resources into the examination of the past, would be the creation of values by an enhancement of present decision making. In a well functioning market economy, which allocates scarce resources in the most efficient way, the mere (medium-term) existence of resources allocated for time series analysis would proof sufficient to state that time series analysis improves current decision quality. If we look at our current economy, there are a lot of use cases for time series analysis. Here are a few examples:

- A technical analyst tries to find patterns within stock market data to predict future price behavior.
- A physician analyzes fever curves to determine therapeutic measures to optimize the convalescence process.
- A sales manager analyzes running sales figures to find out seasonal fluctuations in a market to optimize future production volumes.
- A marketing manager analyzes the search volume for an online shop website to measure success of a marketing campaign.

It can be observed, that time series analysis is done in the context of an array of different objectives. According to Chatfield (2003), the objectives regarding time series analysis may be classified as description, explanation, prediction and control. The following paragraphs briefly summarize all four classes. It has to be noted that the boundaries between them are blurred

Description

For given plot data, the first analysis step usually is to plot the data against time and then obtain simple descriptive measures of the main attributes of the time series.

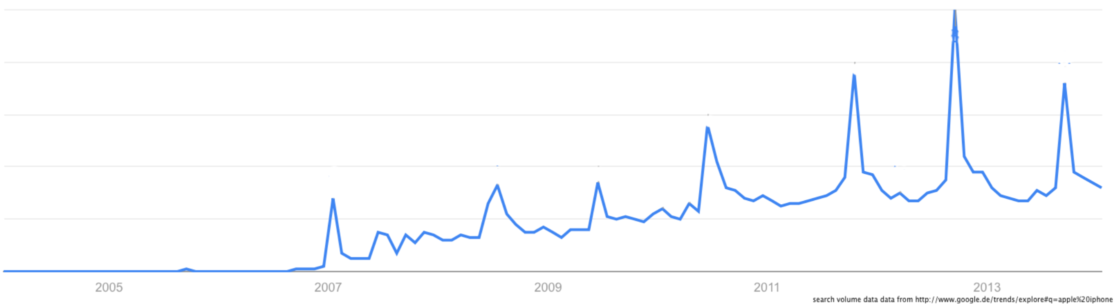


Figure 6 – Univeriate plot of the search volume for the term "Apple iPhone"

The power of a time plot is visible in Figure 6. The graphic of the search volume for the term "Apple iPhone" clearly shows that there is a seasonal effect and an overall upward trend.

If we add an additional data dimension, our analysis can be extended. Figure 7 shows the extended time plot. In addition to the search volume data, this plot also presents information about the date of new product presentations.

With the additional data dimension, it is now possible to make educated guesses about the reason of the seasonality effect appearing in the search volumes.

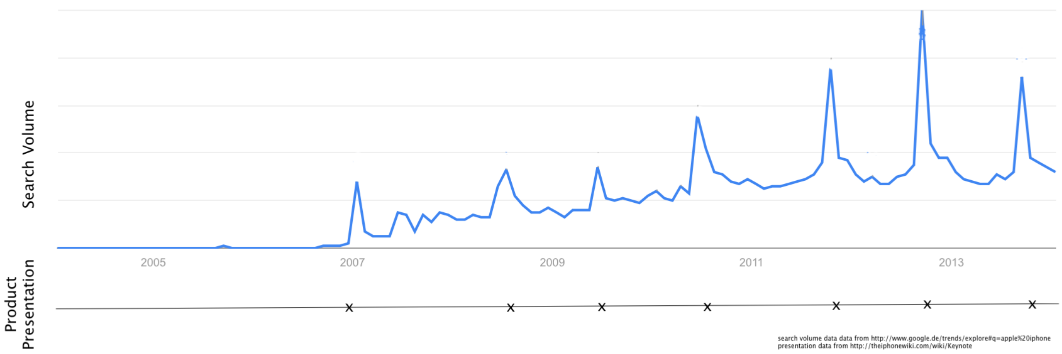


Figure 7 - Multivariate plot of search volume and product presentation data

After studying Figure 7, it becomes very plausible that the seasonal jump in search volume is caused by Apple's product presentations.

Nevertheless, it must also be mentioned that not all time series can be described by such "obvious" features as in the illustration. For these time series it is necessary to apply more complex techniques than visual exploration.

Other features, which are easily spotted in time plots, are sudden changes in behavior of the plot or trend reversal. Furthermore, extreme spikes, which can distort the overall statistic of the data, are easily detected within graphical plots.

Explanation

The extension from a univariate to a multivariate case (as explained before regarding Figure 7) opens up the possibility to use the variation of one variable to explain the variation of another variable. This process may lead to deeper insights regarding the underlying mechanisms that created a particular time series. For example, it is of interest, how a special medication affects the health level of a patient, or how an interest hike affects the investment climate in an economy.

In some cases it is possible to identify such relationships with visual inspection, whereas in other cases, mathematical methods like multivariate regression, neural networks or computerized algorithms are necessary. Generally, these models seek to optimally fill the "model box"', illustrated in Figure 8. Optimally fill in this context means that the model approximates the time series with the given input variables as best as possible (i.e. minimizes some error function)

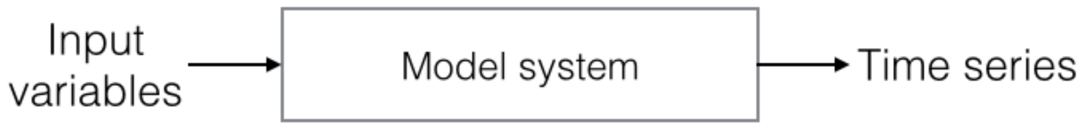


Figure 8 - Schematic representation of the model system

Prediction

Given recorded longitudinal data, one may want to predict future data values of the series. This process can also be realized with the models cited in the previous section "Explanation".

Predication is an important task in sales forecasting, booking behavior estimation or the analysis of economic and industrial time series.

Control

Time series can be collected to improve control over a process. For example, a physician wants to keep the health status of a patient between upper and lower boundary. In many cases, such control problems are closely related to prediction problems. For instance, the physicians will try to predict the future movement of the health status, to be able to adjust medication before the process moves above upper or lower boundary.

2.1.2. Transformation of time series

In some cases it is necessary to transform the raw time series data before plotting. The transformation can be made – for instance – by using logarithms or square roots. Chatfield (2003) cites three cases, in which it is necessary to transform data before plotting:

Stabilize the variance

If there is a trend in the time series and the variance increases with the mean, it is recommendable to transform the raw data. This is particular true if the standard derivation is directly proportional to the mean. In such cases, a logarithmic transformation should be applied.

Create additive seasonal effects

If the size of seasonal effects seems to increase with the mean, it is recommended to make a transformation to keep the seasonal effect constant from one year to another. As before, this is particular true if the seasonal effect is directly proportional to the mean. In this case, a logarithmic transformation is appropriate.

Make the data normally distributed

If the data is not normally distributed, a Box-Cox transformation can be applied to convert data to normality. The Box-Cox transformation for a time series {x} and a transformation parameter λ, is given by:

$$y_t = \begin{cases} (x_t^{\lambda} - 1) & \lambda \neq 0 \\ \log x_t & \lambda = 0 \end{cases}$$

The value for λ should first be guessed, and refined recursively afterwards. Furthermore, it should be noted that sometimes it can be difficult to fully remove non-normality from a data set.

Generally, transformations should be avoided whenever possible. In many cases, it is more difficult to interpret time series and forecasts on transformed data than on the original data set. Furthermore, transformation can introduce biasing effects. According to Chatfield (2003), transformations only make sense if the transformed variable has direct physical interpretation. For instance, when percentage increases are of interest, a logarithmic transformation makes sense.

2.1.3. On missing values

The topic of missing data is particular important in the field of longitudinal studies, where data is recorded from multiple subjects over a specified timeframe. Missing data means that no data values are stored for the variable in a current observation (Nakai and Ke, 2011).

In the case of multi-subject panel data, the problem arises frequently by early drop-out of some individuals. Therefore, the data record for these individuals terminates prematurely. This makes the whole data set unbalanced (i.e. there are more records available at the beginning than at the end of the time series).

For single-subject time series, missing values arise when multiple variables are recorded over time and some variables are missing for some observations. For instance, if a physician monitors a patient's blood pressure, cholesterol and blood glucose over time, an observation with missing values arises if during one appointment only the values for blood pressure and cholesterol are recorded. An example is shown in Figure 9.

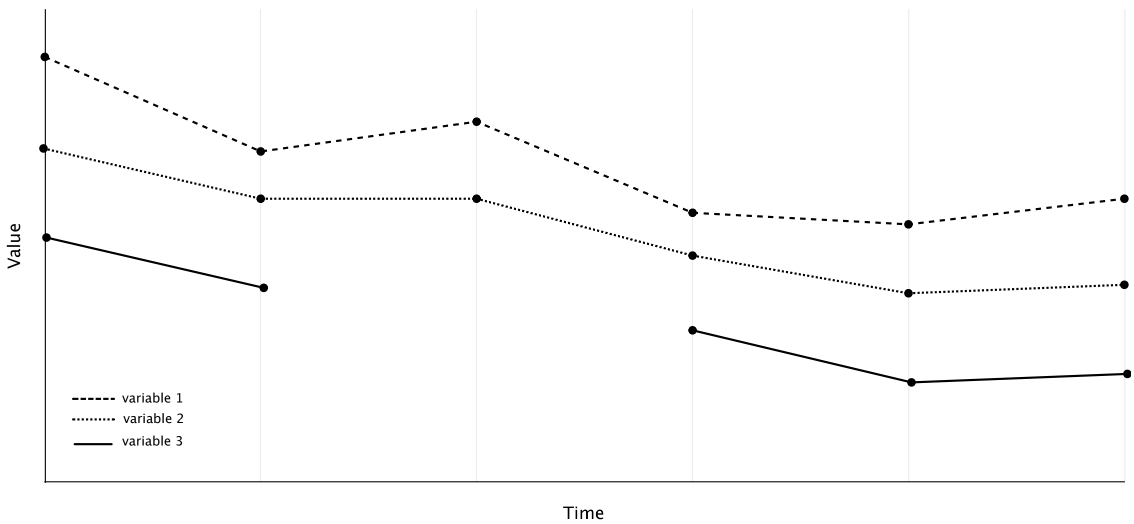


Figure 9 - Problem of missing values

In this case, six observations were made for variables 1 and 2, whereas only 5 observations are recorded for variable 3. The incomplete record was created at the third observation.

Harris (1999) identifies several plotting methods for calling attention to missing data:

- The presence or absence of a plot symbol can indicate missing data (this works only, if the data is recorded with constant frequency).
- Dashed lines can indicate gaps with missing data.
- Gaps within curves can be used to indicate missing data (see Figure 9)
- Different shading color or fill color can indicate missing data.
- No shading color or fill color under missing data makes a bold statement that data is missing (given that the other areas are filled).

Little and Rubin (1987) developed a taxonomy for describing the assumptions concerning the missingness of values. This taxonomy is particularly important if the recorded data is processed by statistical means.

In general, they differentiate between three different classes. Depending on the determined class, an adequate method for statistical handling of the missing values can be chosen:

Missing at Random (MAR)

This means that the probability missing records depends on the set of observed responses, but is not related to the specific missing values that should have been recorded.

Missing completely at Random (MCAR)

Data is said to be missing completely at random (MCAR), if the probability that records are missing is both independent from the specific missing values and also independent from the set of observed responses.

Not Missing at Random (NMAR)

Data is said to be not missing at random (NMAR), if the probability that records are missing depends on both the specific missing values and also the set of observed responses.

It should be mentioned that in reality it is usually very difficult to distinguish between these category classes (Nakai/Ke, 2011). Nevertheless, the following section describes the most important methods to deal with missing data. The method overview is structured using the categories introduced above.

Complete Case Analysis (CCA)

In this approach, only complete cases are used for analysis. The main advantage of this method is that it can be used for any kind of statistical analysis and no special computational methods are required. However, it is necessary that the data is MCAR. Otherwise, the method may lead to a biased outcome because then the complete cases might be unrepresentative for the full population (Nakai and Ke, 2011). This method can be very effective if the discarded missing data is only a small part of the overall sample.

Available Case Analysis

This is a general term for an array of methods that use the available data to estimate a mean and covariance. The most popular method of Available Case Analysis is the pairwise deletion method. In this method, covariance and means are only calculated on time points where a complete data sample is available.

Like CCA, it also leads to biased results unless the data is MCAR.

Single Imputation

This method tries to replace the incomplete observation with information based on an estimate of the unobserved variable's true.

The most common single imputation method is Last Observation Carried Forward (LOCF). In this method, every missing value is replaced by the last observed value of the same subject. Last observation missing is frequently used in the pharmaceutical industry, in the analysis of randomized parallel group trials with the main objective to test the null hypothesis of no difference between treatment groups (Nakai/Ke, 2011).

In many cases, the assumption that data remains unchanged after missing seems unrealistic. Nakai and Ke (2011) mention that one of the few settings, where this assumption might be appropriate is when missing data is due to recovery or cure.

Also, this method requires data to be MCAR.

Multiple Imputation

In this approach, missing values are replaced by multiple acceptable values, which represent a distribution of probabilities. One advantage of this method is that the inferences, such as standard error or p-value obtained from MI, are generally valid because they incorporate uncertainty due to missing values (Nakai/Ke, 2011).

Methods for NMAR data

When data is missing NMAR, standard statistical models are not valid anymore and yield badly biased results.

Methods for handling this kind of data are selection model and pattern-mixture model.

2.1.4. Plotting

"Modern data graphics can do much more than simply substitute for small statistical tables. At their best, graphics are instruments for reasoning about quantitative information. Often the most effective way to describe, explore, and summarize a set of numbers – even a very large set – is to look at pictures of those numbers. Furthermore, of all methods for analyzing and communicating statistical information, well-designed data graphics are usually the simplest and at the same time the most powerful"

-- (Tufte et al., 1983).

As Tufte et al. (1983) describe, the right visualization can dramatically improve the data exploration process.

Nevertheless, first time series visualizations in scientific writings have not been established before 1750-1800. William Playfair (1759-1832) developed or improved almost all of the fundamental graphic designs with the aim of replacing conventional tables of numbers with the systematic visual representations of his "linear arithmetic" (Tufte et al., 1983).

Nowadays, the most common visualization techniques for longitudinal data are point charts, bar charts, line graphs, sequence graphs and circle graphs (Weber et al., 2001). In the following paragraphs, these techniques are briefly introduced.

Sequence graphs represent time-dependent data on one dimension by indicating each data point with a mark on the axis. The distance between each mark on the axis represents the time span passing between the events. With sequence charts, it is not possible to visualize a second dimension for a data point.
Figure 10 shows a sequence graph. In this example the graph visualizes treatment frequency information.

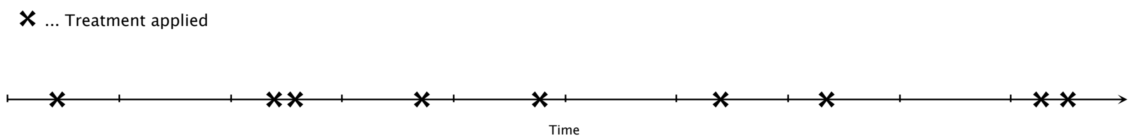


Figure 10 - Sequence graph

This mode of presentation allows an easy recognition of treatment pilings and longer time frames without treatment. Nevertheless, it is not possible to add additional information, like a dosage quantity related to the treatment, to this visualization.

Therefore**, point graphs** extend sequence charts by using a second axis to display a further information dimension. The distance from the main axis thereby represents the second data dimension. Figure 11 shows a point chart. In this example the point chart visualizes treatment frequency and quantity information.

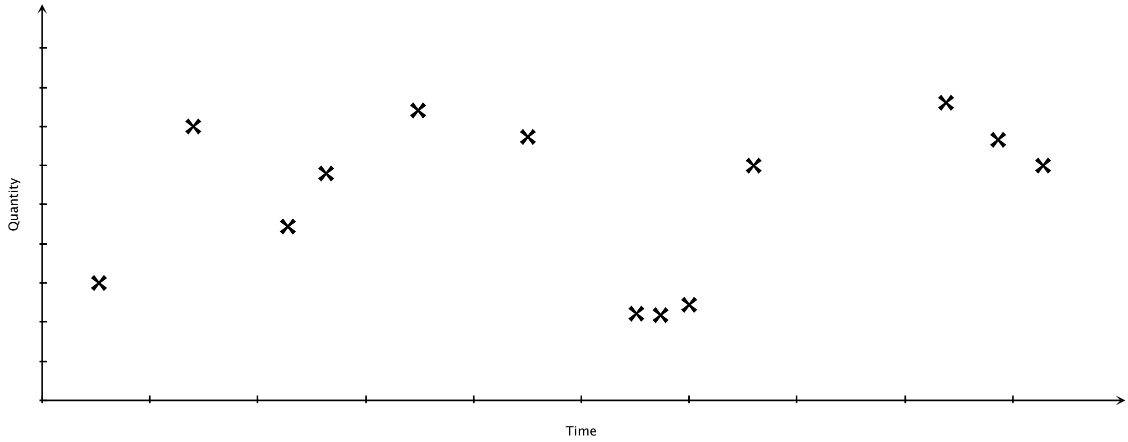


Figure 11 - Point graph

The vertical distance to the origin represents the quantity information, whereas the horizontal distance to the origin stands for the elapsed time.

Bar graphs replace the points with bars, which the comparability between data points. Figure 12 shows the same information as represented in Figure 11, visualized as a bar chart. A comparison of these two visualizations shows that the ability to compare data point quantities is enhanced by using bar charts.

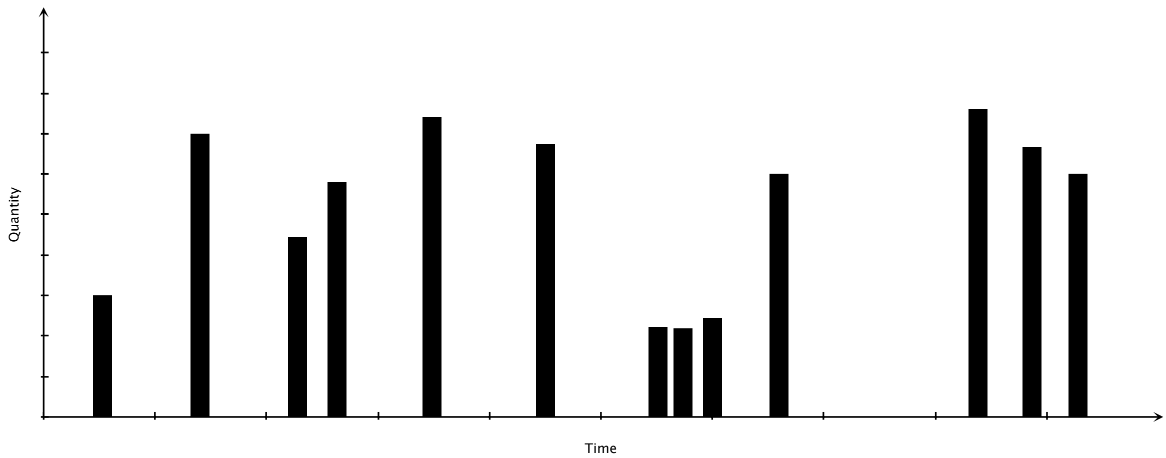


Figure 12 - Bar graph

Line graphs extend point charts by connecting the dots with lines to emphasize the temporal aspect of data. Line charts are very helpful for indicating trends over time.

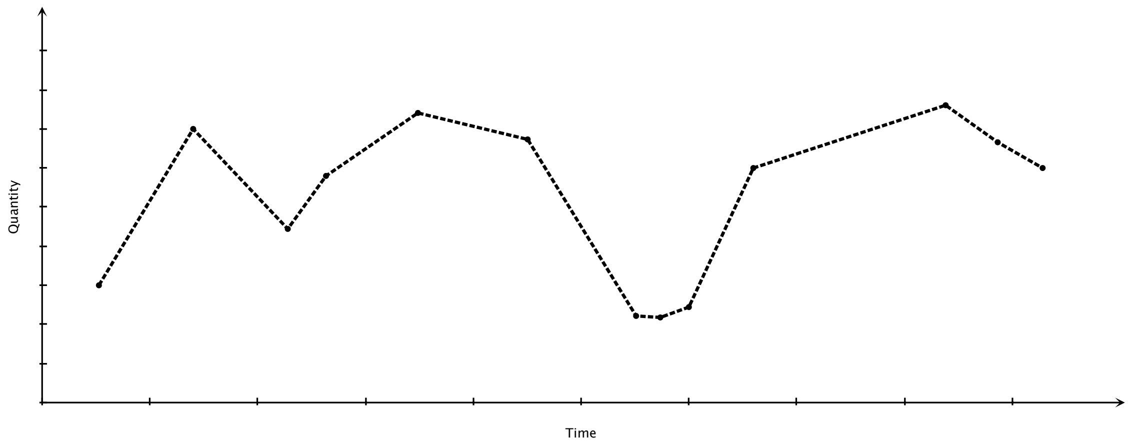


Figure 13 - Line graph

Circle graphs map the time series data into a spherical domain. They are commonly used to visualize periodic data with a known cycle length. Figure 14 shows an example of a circle graph, in which a fictive average internet usage time of a user is plotted as circle graph. As is shown in the picture, the are within the data line is usually filled in circle graphs (Harris, 1999). The shape of the area makes different periodic behavior comparable.

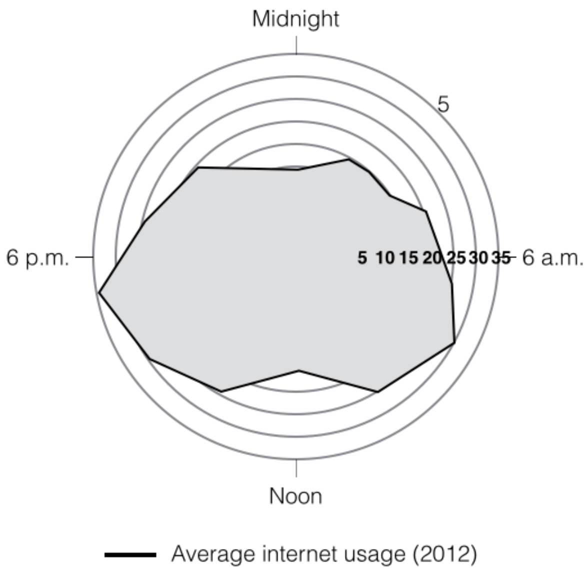


Figure 14 - Circle graph

Weber et al. (2001) introduced a **spiral graph** to visualize evenly spaced longitudinal data. Instead of plotting it by using a bar or line chart, they used a spiral information visualization. This method enhances the ability of users to visually identify recurring patterns within the dataset.

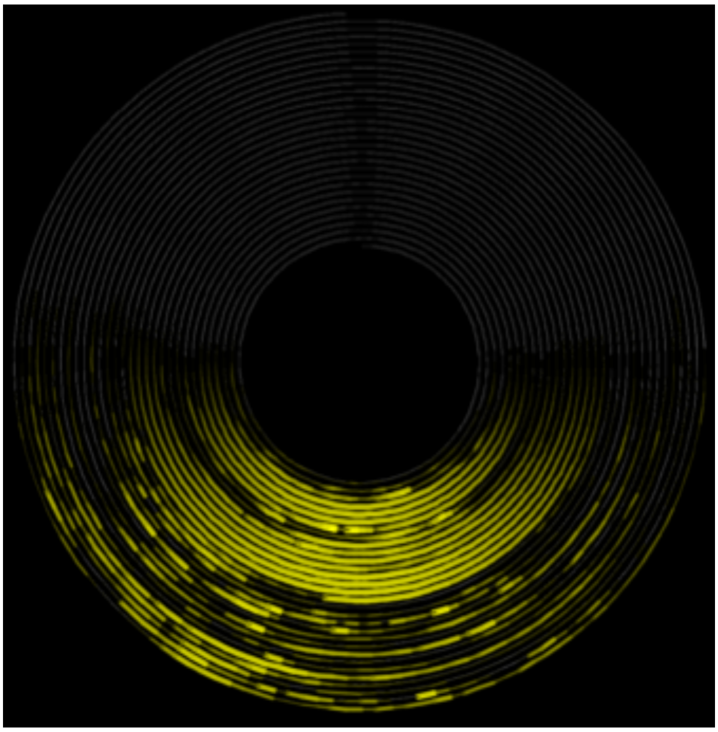


Figure 15 - Spiral visualization of sunshine intensity (source: Weber et al., 2001)

Figure 15 shows an example of spiral visualization, which encodes the sun intensity with the color of the spiral line. In the diagram, it is very easy to detect day and night cycles as well as cloudy periods within one day.

The aspect that multiple sequences can be merged into one chart to allow for comparisons.

Additionally, if multiple sequences add up to a total value sequence, which is as important as an individual sequence, it is possible to visualize the data with a **stacked line graph**. This stacked type of visualization is also often used to display evolution of shares (e.g. market shares) over time. Although this second kind of use case for stacked-line graphs shows the evolution of shares in clearer way, it has the drawback that information about the evolution of absolute values is lost.

The differences between "share" and "absolute" stacked graphs are shown in Figure 16 and Figure 17. In both, the same underlying data set is displayed. First we use a "share"-type stacked graph and in the second example an "absolute"-type stacked graph. Figure 16 shows of "share"-type visualization. In the example, the market share distribution of a fictional market is displayed. The values at each point of time add up to a total of 100 percent. A hasty interpretation of this figure would lead to the conclusion that only company A has experienced substantial growth, whereas company B has shrunk substantially and company C has kept almost the same size.

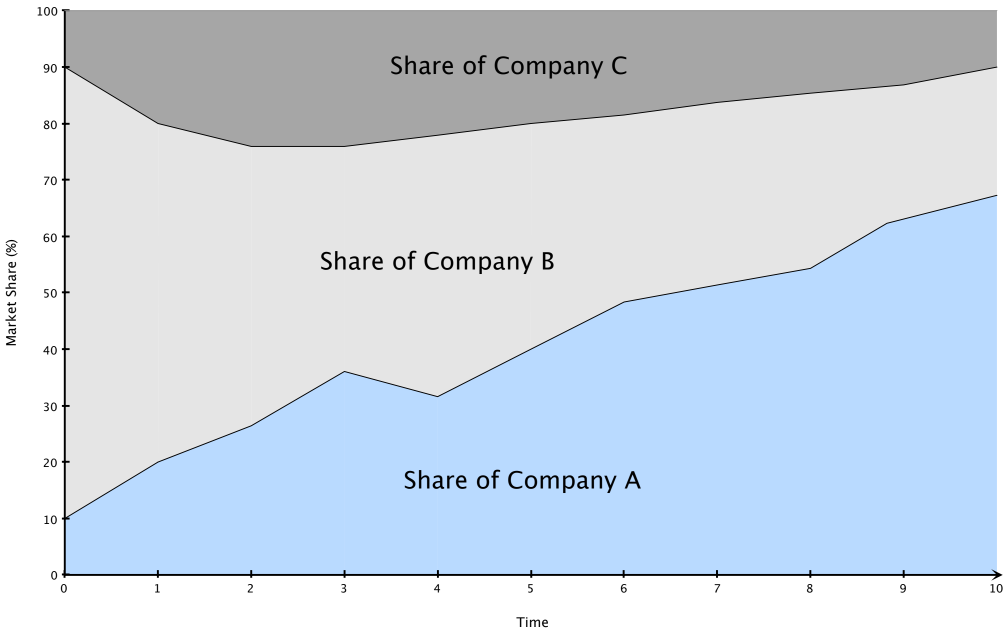


Figure 16 - Stacked line graph for displaying market shares

This impression is not quite true if Figure 17 is taken into account as well. It shows that, although company A experienced by far the strongest growth, the overall market grew very rapidly from 300 to 900 units too. Additionally, the figure shows that company B has not experienced a decline in absolute unit terms.

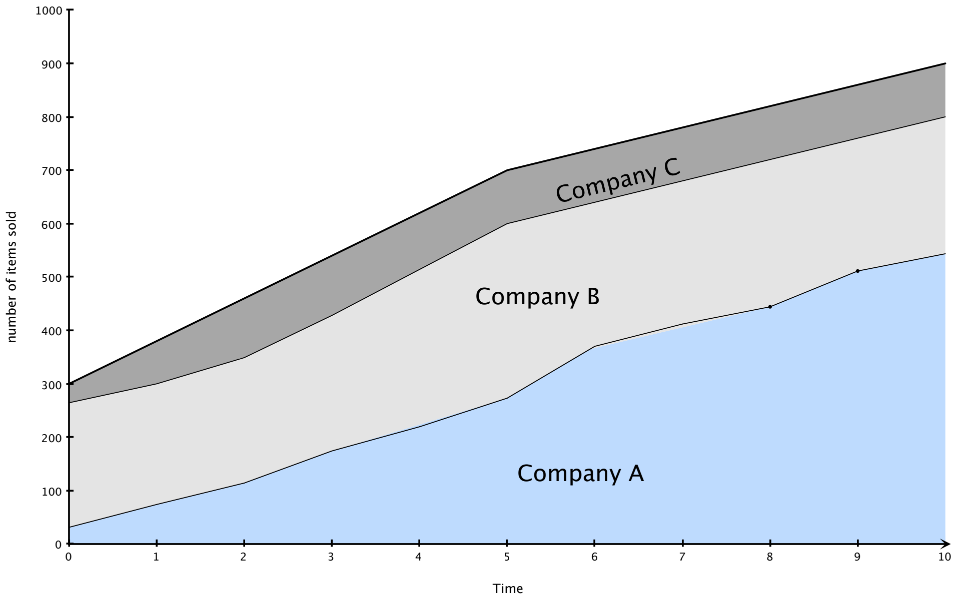


Figure 17 - Stacked line graph for displaying absolute growth

This example should illustrate that a simple change of data visualization (and also a simple transformation of underlying data) can lead to a huge difference in interpretation. Furthermore, by adding 3D visualization, line charts can be extended by another dimension.

According to Tufte et al. (1983), time series graphics are most suitable for big, complex data sets with real variability. Simple linear changes should better be summarized with one or two numbers.

Additionally, he introduced the following common guidelines that should be considered by every graphical display:

- show the data

- induce the viewer to think about the substance, rather than about methodology, graphic design, the technology of graphic production, or something else
- avoid distorting what the data should express
- present many numbers in a small space
- make large data sets coherent
- encourage the eye to compare different pieces of data
- reveal the data at several levels of detail – from broad overview to fine structure
- serve a reasonably clear purpose: description, exploration, tabulation, or decoration.
- be closely integrated with the statistical and verbal descriptions of the data set.

Aris et al. (2005) discuss challenges for representing unevenly-spaced time series data (event-based time series). Unevenly-spaced means that the time span between measurement points can vary in the dataset. They present four methods to deal with this kind of data:

- **Sampled events** – for this method, y-values of the data are sampled at regular predefined intervals. Sampling requires an algorithm to determine which y-value to use when no data point is available at the time of sampling.
- **Aggregated sampled events** – this method extends sampled events by allowing the user to aggregate samples to generate zoomed versions with more/less samples. An aggregation mechanism for joining surrounding sample points is required. An obvious example would be to take the mean of the aggregated points.
- **Event index** – instead of representing time linearly on the x-axis, this method plots data according to the index of appearance with constant spacing between events.
- **Interleaved event index** – this method extends the event index by spanning it over all visualized time series. All events are shown in the order of appearance but the spacing between events does not represent the duration between events.

2.1.5. Plotting Range

At times, it can be necessary to visualize not only a single point, but a range of values for each observation. This can be the case, if the observed data points comprise some margin of error or if multiple observations are compressed into one time point of the plot. The latter is often done with stock market data. As stock market prices vary over a whole trading day, valuable information would be lost if only the closing price was plotted on a daily basis. Therefore, not just closing, but day-open, day-close, day-high and day-low are visualized.

In the following paragraph some techniques for visualizing such range information will be introduced. Although many of the techniques are most frequently used with stock-market data, they can easily be applied in other areas of use.

Figure 18 shows three common visualizations for range information. The first way of introducing range into a plot is the incorporation of **error bars**. The horizontal bar symbols indicate some lower and upper threshold for each value. Error bars provide no possibility to introduce more information, like day-start and -end values, into the visualization.

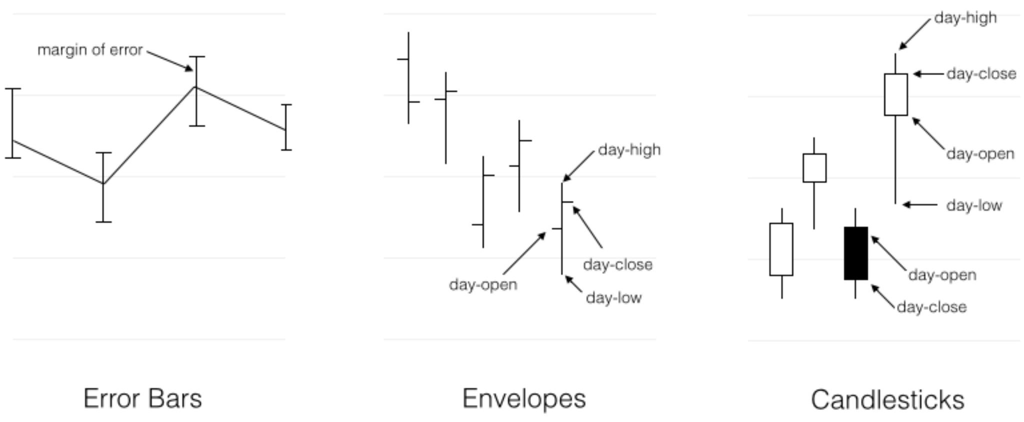


Figure 18 - Different techniques to incorporate range information

A second way to include range information is by using **envelopes**. As shown in Figure 18, envelopes provide information about the range as well as day-start and -end values.

The same kind of information can be visualized with **candlesticks**. Whereas envelopes encode the information about start and end values by adding a small line to the left and right of the range line, candlesticks use the coloring of a rectangle. Black coloring means that the top line of the rectangle is the start- and the bottom line is the end-value. White coloring inverts this convention.

Figure 19 shows a box plot visualization. Box plots can provide information about the basic distribution properties of the data. The bottom and top of the box are first and third quartiles, and the band inside the box is the median. The ends of the lines, extending the box, are called whiskers. They can represent several possible alternative values. In this example, they stand for the minimum and maximum of the data set.

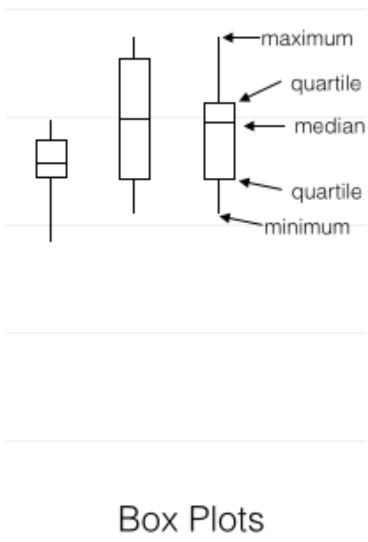


Figure 19 - Box plot

2.1.6. Aspect Ratio

One crucial parameter when creating a time series plot is the aspect ratio, defined as $\alpha = \frac{height}{width}$ (Bisgaard/Kulahci, 2011). This parameter has a significant impact on the viewer's ability to detect slopes.

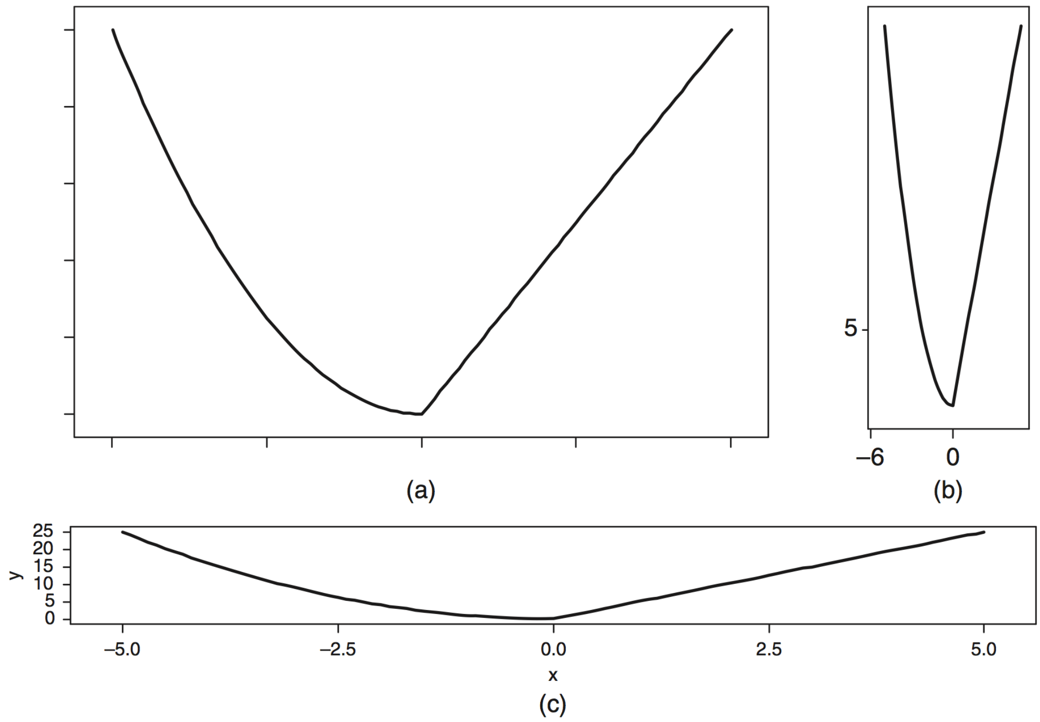


Figure 20 - Same data plotted with different aspect ratios (source: Bisgaard/Kulhaci, 2011)

Figure 20 illustrated the problem quite impressively. In this graphic, the same data points are plotted with different aspect ratios. The differences between the two curve segments are best seen in version (a) of the plot. This result is based on the biological fact that our eye has the best discriminative power when the orientation of what we like to compare is close to 45 degree. Therefore, the aspect ratio should be arranged in such a way, that objects, which should stand out, have an average ratio around 45 degrees.

Overall, it can be quite beneficial to try different aspect ratio before deciding which one to use.

2.1.7. Chartjunk

Chartjunk (Tufte et al., 1983) refers to all visual elements within charts and information graphics that are not necessary to convey any information represented on the graph. Instead, they distract the viewer from this information.

Elements can be termed “chartjunk” if they are not part of the minimum set of symbols to communicate information understandable way. Well-designed charts should avoid these elements, to draw the viewer’s attention to the meaning and substance of the data. This can be achieved by maximizing the data/ink ratio of a graphic.

Tufte et al. (1983) define the data ink as the minimum, non-erasable core of a graphic, the non-redundant ink arranged in response to variation in the numbers represented.

$$\text{Data/ink ratio} = \frac{data\text{-}ink}{total\ ink\ used\ to\ print\ the\ graphic}$$

This means, the data/ink ratio corresponds to the proportion of a data graphic’s ink devoted to the non-redundant display of data-information.

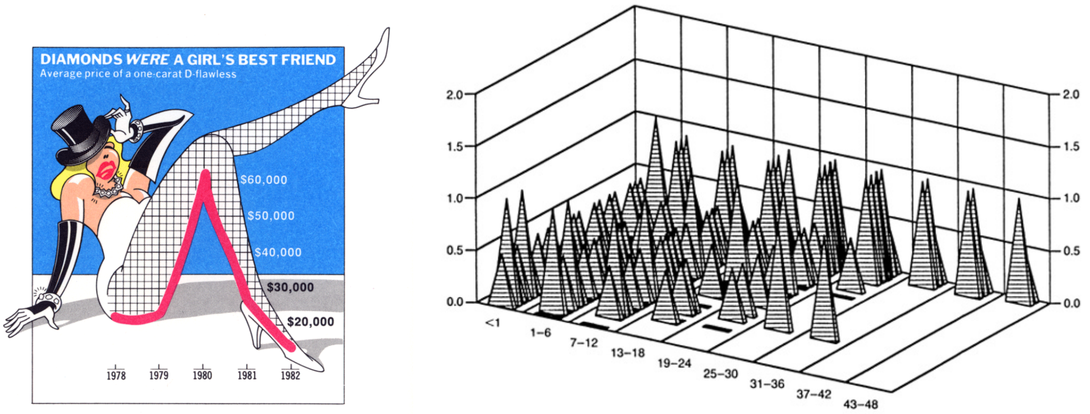


Figure 21 - Examples for chartjunk (source: Tufte et al., 1983)

Figure 21 shows two example data graphics with an excessive amount of chartjunk. Both graphics have a very low data/ink ratio. On the left, a lot of eye-catching optical art is used, which distracts from the actual information.

The graphic on the right includes distracting appearances of vibration and movement, called moiré effects. These effects occur when design interacts with the physiological tremor of the eye. Furthermore, some pyramids conceal each other and the depth of the graphic has no label or scale.

2.1.8. Interaction with time series

Datasets are sometimes too huge to fit every data point in a reasonable way on one page or screen. There are different approaches to tackle this problem.
One common method is to reduce the underlying dataset size. This can be reached by downsampling the data. For instance, a set of around 30 million data points, each representing a second, could be down-sampled to 8760 data points by averaging to a one-hour representation. For the representation of the downsampled data points, visual elements introduced in chapter 0 (plotting range) can be used. To give an example, it is common practice to visualize aggregated stock market data by using candlesticks or envelope visualizations.
Additionally, Hao et al. (2007) address the problem of visualizing large time-related data sets on devices with limited screen size. In their approach, they allocate screen space in proportion to the degree of interest for data subintervals. The degree of interest (DOI) for data intervals is either preconfigured for a particular application domain or derived from the data by a suitable algorithm.

Electronic devices can solve this problem by allowing users to interact with the displayed data in an explorative way. The device displays only a subpart of the data which is small enough to fit on the screen, and supports navigation between different regions of data to see the full picture.
Additionally, this technique is often combined with zooming in and out, which implies the already mentioned downsampling of data.

According to Weber et al. (2001), the following interaction methods can enhance the information perception process with these graph types.

- *Zooming* – initially, a high level overview of the course of the time series is given. By zooming, the user can obtain a more detailed view on a subarea of interest.

- *Scrolling* – if the area of display is not large enough to fit the whole chart (e.g. after zooming in), the user can scroll through areas.
- *Focusing and linking* – extends the idea of zooming by providing not only a zoomed-in version of the data, but also applying different, more effective visualization techniques to the zoomed-in dataset.
- *Brushing* – provides the idea of extended data visualization by automatically displaying pop-ups as a roll-over effect.
- *Filtering* – taking away (ignoring) irrelevant data objects.

In chapter 0 “Multi-touch interaction” we will explore, whether multi-touch displays have additional benefits in comparison to mouse-based devices, for implementing these methods.

2.1.9. Dynamic time series visualization

Dynamic time series visualization typically displays multivariate data as a series of time slices. A time slice thereby encodes the structure of the series at a given time (Archambault, 2011).

The majority of dynamic drawing algorithms use animation to display the sequence of time-slices. In an animation, the positions of nodes are interpolated between two time slices. Nodes that have been added or removed are faded in/out of the screen. A famous example for animated drawing is implemented in the Gapminder software (http://www.gapminder.org) from Hans Rosling (Rosling et al., 2006). Gapminder is a software package to display development statistics in an interactive, animated way. Figure 22 shows on overview of the Gapminder functionality. By using y-axis, x-axis, size and color of the bubbles, four information dimensions can be presented at the same time. The development of the data values in the course of time is visualized by animation of the bubbles.

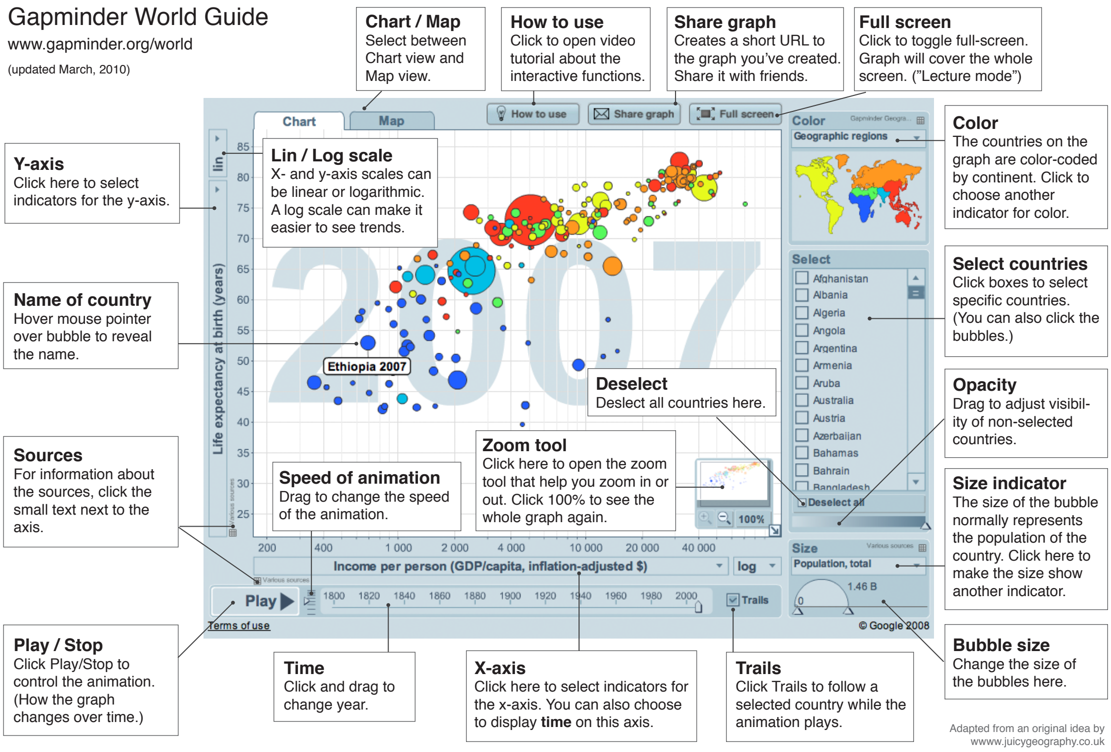


Figure 22 - Guide to the Gapminder software

According to Archambault (2011), animated visualization brings benefits for mental clustering of nodes. Objects that move similarly, tend to be mentally grouped together, even if they are actually very far apart. This is the case because similarly moving objects in a drawing create a pop-out effect from a large and potentially cluttered background.

2.1.10. Longitudinal data in health care

As the course of a patient's disease unfolds over time, longitudinal data naturally plays a very import role in health care (Musen/Helder, 1997). Together with this accumulation of disease data the physician's insight may evolve over time.

Also, patient records are chronological accounts of observations, interpretations and interventions. This time-oriented data is essential for detecting trends when following the condition of chronically ill patients. The physician relies on this data for decision making, such as continuing a prescription or repeating a test.

Furthermore, longitudinal data is starting to play an important role in the personal health and fitness sector. Especially the ongoing trend to track health information with smart sensors, creates a huge amount of longitudinal data. This trend can be summarized under the term “quantified self”. Swan (2009) defines quantified self-tracking as the regular collection of any data that can be measured about a person. This includes biological, physical, behavioral and environmental data. Ultimately, these large amounts of self-tracked data need to be converted into something human-readable. In most of the cases, this conversion implies longitudinal data visualization.

2.1.11. Frameworks for visualization

There is a variety of frameworks for conducting data visualizations, most of them freely available in the web.

Visual.ly (http://visual.ly) claims to be the world’s marketplace for visual information. It is a community platform, where users are encouraged to create visualization from their personal data and share it with others.

Another visualization tool available on the web, is the IBM Many Eyes platform (http://www-958.ibm.com/software/analytics/manyeyes/visualizations). The website allows to participate by uploading new datasets or by creating new visualizations with new or existing data sets. Furthermore, it is possible to comment on all existing datasets and visualizations. Regarding visualization options, an array of different types is supported. For time series, they explicitly support line graphs and stacked graphs.

Fluxstream (https://fluxtream.org) is an open-source, non-profit personal data visualization framework. It is specifically aimed at creating visualizations from multi-stream sensor inputs. The platform supports data import from several popular tracking services (e.g. RunKeeper, Moves). Its mission is to help users make sense of their life (i.e. the tracked sensor data) and to help them test hypotheses about what affects their well-being.

Google Fusion Tables (http://www.google.com/drive/apps.html#fusiontables) is a framework that provides data management and visualization services (Gonzalez et al., 2010). The set of available visualizations is computed-based on the data types found in a fusion table. For longitudinal data, plotting options from Google Charts (https://developers.google.com/chart/) can be used. Google Charts is an API that lets developers create charts from data sources and include it into a webpage.

2.2. Multi-touch interaction

This chapter gives a brief overview of the history of multi-touch devices and afterwards discusses the interaction difference between multi-touch input and mouse-based input. Furthermore, some use cases of multi-touch devices within the healthcare sector are presented.

2.2.1. Definition and History of Multi-touch input

Referring to the patent US7,812,826B2 (Ording et al., 2006) for a portable electronic device with multi-touch input, a multi-touch device detects one or more multiple-touch contacts and performs one or more actions based on the contacts. Figure 23 displays the process flow chart for such kinds of multi-touch interaction. The flow chart shows that interaction is based on motionless display contacts (e.g. touching a displayed button) as well as display contacts, which include motion (e.g. a swipe from left to right). Furthermore, box 204 in Figure 23 shows that the chosen action also depends on the number of screen contacts.

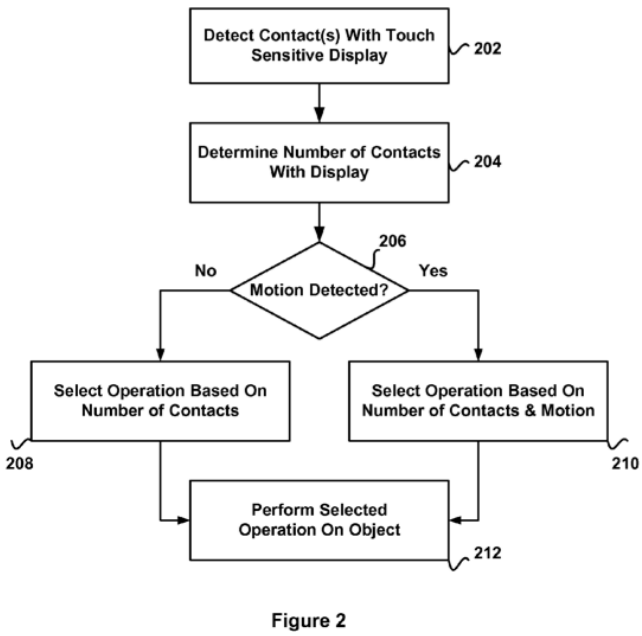


Figure 23 - Process Flow Chart for multi-touch interaction (source: Ording et al., 2006)

Although Mitsubishi developed a first multi-touch screen in 2001, it took another six years longer for this new device category to gain wider adoption. In 2007, Microsoft introduced the Microsoft Surface tabletop computer (rebranded in 2012 to Microsoft PixelSense after the launch of the Microsoft Surface tablet lineup), which got considerable media attention. In the same year, on June 29, Apple introduced the iPhone. Together with the iPhone launch, devices with multi-touch display started to gain noteworthy market shares within the computing sector. In 2010, Apple introduced the iPad, a 9.7 inch tablet computer based on the same multi-touch technology as the iPhone. This launch accelerated the trend even further. Within a short time after their introduction, multi-touch devices have entered various areas of life and have been adopted as tools for many different use cases.

This is reflected in the growing market penetration of tablet devices. According to a Gartner survey, the shift from conventional desktop machines to tablet devices will gain additional speed in the coming years. Whereas annual PC sales decrease from about 341 million to 272 million between 2012 and 2017, tablet sales are going to increase from 116 million to 468 million (see Figure 24).

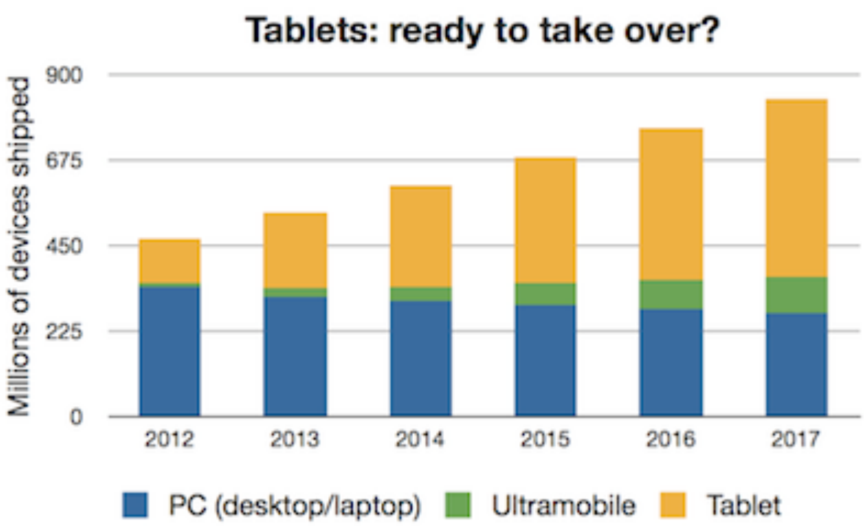


Figure 24 - Estimated sales figures for PC, table and ultra-mobile devices (source: Gartner, http://www.gartner.com/newsroom/id/2408515, graphic by The Guardian)

Although multi-touch technology has experienced this steep rise in adoption, it should not be seen as a replacement of conventional desktop, mouse-based technology. Depending on the concrete tasks, both device categories can have their benefits. In the following section, performance of multi-touch and mouse-based devices will be compared.

2.2.2. Multi-touch vs. mouse-based devices

Compared to mouse-based devices, multi-touch devices provide a greater richness of interaction possibilities. In this case, richness of interaction refers to the degrees of freedom in interaction supported by the technology (Buxton, 2007).

Conventional mouse-based interfaces (the WIMP – Windows Icons Menu Pointers) rely heavily on a single 2D-cursor, which results in 2 degrees of freedom (not counting the state of the mouse-button). Sensing multiple fingers on a multi-touch display, however, results in a multiplication of the degrees of freedom. Including the various other sensors, that come with most of the multi-touch devices, like gyroscope, accelerometer, microphone or camera, further expand interaction possibilities. All these sensors enable an interface designer to capture more about a user's state and include this information into the interaction concept. Unfortunately, many UIs do not take full advantage of technologies. Instead, it seems that they are designed in a "lowest common denominator" manner, to be applicable to many different devices at once without much redesigning effort. These UIs are not tightly coupled to a device's hardware properties. Wigdor and Wixon (2011) assess that a good UI design should be tailored to device properties and not be implemented in a generic way.

According to Buxton (2007), appropriate use of the technological advantage has the potential to beginning to capture the type of input richness that we encounter in everyday life.

Precondition to capture the benefits of the greater degree of freedom in interaction is a set of established gestural interaction guidelines. A well-designed gestural interface can shorten the learning curve by replacing a maze of menus and controls with simple actions, gestures, affordances and feedback (Wigdor/Wixon, 2011). Nevertheless, gestures must be self-revealing, memorable and associable with a task and, furthermore.

The UI should react immediately and continuously to a gestural command. Norman and Nielsen (2010) criticize some of the current trends to that effect:

- The lack of established guidelines for gestural control
- The misguided insistence of companies (e.g. Apple and Google) to ignore established conventions and establish ill-conceived new ones.
- The developer community's apparent ignorance of the long history and many findings of HCI research, which results in their feeling empowered to unleash untested and unproven creative efforts upon the unwitting public.

Nowadays, there are already some de facto gesture standards established for common tasks. The following list gives an overview of some popular gestures and associated tasks:

- *pinch gesture* – continuous zooming-out
- *spread gesture* – continuous zooming-in
- *double-tap gesture* – discrete zooming-in and discrete zooming-out (overview / detail)
- *one-finger swipe* – changing pages back / forth
- *two-finger swipe* – changing between applications
- *long-press gesture* – opening context menu

Nevertheless, the question of how visualizations need to be adapted and redesigned to maximize the benefits of this greater freedom in interaction, is still a largely open research direction. Isenberg et al. (2013) present a systematic overview of this issue.

Viewed from a more low-level task side, user tests have yielded the solid result that high-precision tasks are performed faster on a mouse-based device, whereas complex mouse asks (e.g. rotating a object and scaling) are faster done with multi-touch (Muller, 2009). In the following paragraph some user studies regarding touch-performance will be summarized.

Muller (2009) compared mouse and multi-touch input by tracking the performance of test participants on tasks that focus on pointing and selection, gesture-based interaction and collaborative problem solving. Based on the experiment's results, three main findings were formulated:

- The performance of a task on a multi-touch device depends on the used hardware. The most important hardware issue is the friction of the multi-touch surface. Low friction improves the user experience.
- Tasks, which require precision, are performed faster with a mouse-device and complex mouse tasks are done more quickly when using a multi-touch device.
- Multi-touch devices encourage collaboration. The test results show significant improvement when the number of users is increased.

Kin et al. (2009) showed in an experimental setting that multiple-target selection with multi-touch is about twice as fast as mouse-based input. The experiment was done by displaying targets and distractors (boxes in different colors) on the screen and asking participants to mark the targets as fast as possible. The results show that direct touch with one-finger accounts for 83 percent of the speedup. The remainder is made up by bimanual interaction. Furthermore, they found a strong tendency within their participant group to favor multi-touch input over the mouse-setting. In an ex-post questionnaire, participants were asked, if they preferred the mouse-based or the multi-touch input device. The question scaling ran from 0 (strong prevalence for mouse-based input) to 9 (strong prevalence for touch based input). The average score for this question was 8.3, which indicates a very height subjective preference for the multi-touch option. Based on the results, Kin et al. (2009) suggest the following design principles for multi-touch workstations:

- Even supporting just one finger input can deliver large performance gains over mouse-based input, especially for multi-target selection.
- Support for detecting two fingers will further improve the multi-target selection performance. More than two finger support is unnecessary to improve selection speed.
- Same hand multi-finger usage should be reserved for gesture usage or controlling multiple degrees of freedom rather than multi-target selection.
- The size of the targets should be at least as large as the size of one fingertip.

Frisch et al. (2009) presented a user study, investigating multi-touch diagram creation, movement and editing. Study participants where asked to perform spontaneous gestures for 14 pre-selected tasks. For interaction, the users were able to choose between three different ways: single-hand interaction, both-hands interaction as well as using hand and pen together. All activity was recorded and enriched with thinking-aloud data. With the data of their user tests, Frisch et al. (2009) established a gesture set for common activities like node creation, zooming, node scaling.

Lee et al. (2012) discussed the importance of "natural" interaction methods like multi-touch input for modern information visualization and knowledge discovery. They identified several underexplored methods of interaction, which can provide a new, more "natural" way of interaction for information visualization. According to the authors, five opportunities and challenges for the information visualization community are:

- *Go beyond mouse and keyboard*. New input methods make it possible to reduce the number of menus and controls needed in a mouse and keyboard interface. The more simplistic user interface should allow the user to focus more on the data.
- *Provide a high degree of flexibility in expression*. This flexibility should support the analyst in expressing and analyzing their research question and hypothesis in a fluid, highly configurable way.
- *Take social aspects into account.* Visualization environments should support collaborative information tasks.
- *Reduce the gap between a person and technology*. This should help to make the tool available to a wider audience and offer more people the chance to explore data visually.
- *Gain a better understanding of people's behavior*. This should help create a better interaction experience.

Walny et al. (2012) investigated pen and touch interaction for data exploration on interactive whiteboards. They criticize the issue of too complex mouse and keyboard interfaces and refer to pen and multi-touch input as one way to make the user input more "natural". Within their user study, five main findings were established:

- *Dedicated tasks for pen and touch*: The participants clearly distinguished between appropriate pen and touch interactions. Some interactions were performed by touch, although the pen was still in their hand.

- *Explorations into new interaction paradigms*: Although predominant menu-based interaction influenced initial interaction, participants readily switched to new interaction methods.
- *Working towards integrated interaction*: Interactions should be located in close proximity to the elements of interest, rather than using externally located interface widgets.

Wallace et al. (2013) investigated the support of digital tables and personal tablets for collaborative sensemaking. Their findings revealed a positive impact on group performance, using digital multi-touch tables and a negative impact using smaller personal tablets.

2.2.3. iPad in Healthcare

Due to the circumstance that multi-touch tablets are quite a young technology, there is still a lot of experimenting in finding use cases. Also in the healthcare sector, there is a high diversification into different areas of application for tablet devices nowadays. Time will show, which of them will prove overall beneficiary in the long run.
The CIO magazine states, inter alia, the following examples for using the iPad in a medical setting[1]:

- *The iPad as a replacement for the clipboard*
  Patients can fill out forms and questionnaires electronically before an appointment, giving physicians real-time access to the collected information.
- *The iPad for engaging patients*
  Some US hospitals offer special iPad Applications for tracking waiting times, finding physicians and storing personal medical information.
- *The iPad supports home health initiatives*
  With built-in camera, network ,Bluetooth 4.0 capabilities and less configuration effort, the iPad may bring home health to the masses.
- *The iPad makes medical software more accessible*
  Much medical software is built on old code, which runs on relatively old

[1] http://www.cio.com/slideshow/detail/78144

operation systems (e.g. Windows XP). The iPad's touch interface forces developers to create new applications, with new interfaces and new codebases.

- *The iPad forces* EHR *vendors to improve usability*
  The US government aims to convert all medical records into electronic form by 2014 and by 2015, penalties for entities dealing with patient healthcare data in non-electronic form are likely to be introduced[2]. Usability shortcomings, including bad interfaces, poor data entry and cluttered visualization, is a significant hurdle to EHR adoption. The native iPad EHR system, built specifically for the tablet's touch interface, should improve usability.
- *iPad improves surgery preparation, recovery*
  Surgeon teams need to access a lot of data before, during and after a procedure. The iPad can support all of these phases. During a procedure, an iPad placed in a sterile sleeve allows review image studies previously saved to the cloud. Furthermore, live monitoring of vital signs, object recognition and videoconferencing are additional use cases for this area.

Nevertheless, there are also a lot of challenges intertwined with the usage of iPads in the healthcare sector. Issues include incompatibility with legacy applications, concerns about device durability and sanitization, impact of wireless networks and security, and risk management.

[2] http://www.myemr360.com/emr-mandate-2014

2.3. Medical Information Systems

The blueprint of a Medical Information System is defined in US Patent US6,332,502 B1 by Schoenberg et al. (1999). The authors define it as a system that receives patient data from various sources and displays it in a variety of different formats for use by members of a medical team in a hospital, clinic or office. Figure 25 shows an exemplary display screen filled with patient data.

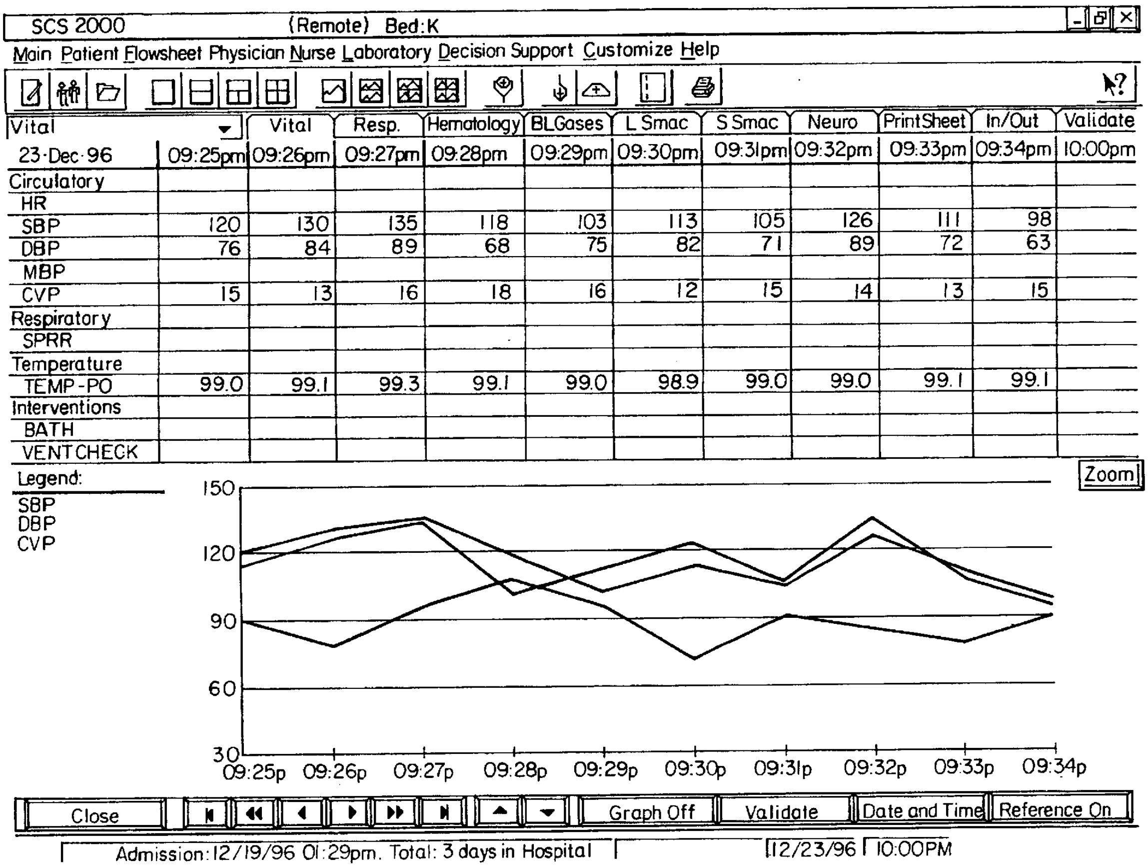


Figure 25 – Exemplary display screen of a medical information system (Schoenberg et al., 1999)

As seen above, on the one hand the system shows detailed information in form of precise numbers in the table on the top of the screen, and on the other hand also a visualization of the trend of selected parameters on the screen bottom.

Schoenberg et al. (1999) define the following objectives for medical information system:

- Display all types of medical information in one place, in a variety of easily understood formats.
- Receive and bundle patient data from a wide variety of sources, including physicians, pharmacists, monitoring equipment, testing laboratories, self-tracking information and other computer databases.
- Gain access to predefined subsets of the data by selecting "job function" icons.
- Provide capability to record observations about a patient, using key words and phrases, which can be supplemented with additional text for customized notation.
- Ability to present multiple types of patient data at the same time.
- Provide real-time data updates.
- Simultaneous access for more than one user.
- Provide graphical display of multiple types of patient data using a common time scale.
- Provide score computation and identification of missing parameters or values for successful computation.
- Provide storage of patient data – at least for the period of current hospitalization.
- Provide recall capabilities for patient data created during previous hospital stays.
- Provide active control of peripheral devices, such as respirators, infusion pumps, intravenous pumps, etc.

Depending on the exact usage circumstances, a medical information system implements one or several of these objectives.

3. Related work

3.1. Rheumatologic Clinical Quality Management System (RCQM)

Introduced by Simonic et al. (2011), RCQM is a clinical quality management system, which supports the therapeutic treatment of patients with rheumatoid arthritis. It helps physicians to tailor the assortment and dosage of drugs for each individual patient. It accomplishes this by calculating a score value, which represents the patient's disease activity and displaying this disease activity score (DAS) as function over time. The resulting DAS patterns and trends can be used to optimize the treatment and to make assertions regarding the quality of the therapy. To calculate the DAS, a scoring algorithm is applied on more than 100 clinical and functional parameters of a patient. RCQM also supports the process of gathering these patient parameters in an efficient way (less than 10 minutes time needed).

Subsequently, rheumatoid arthritis and RCQM are described in more detail.

3.1.1. Rheumatoid arthritis

Rheumatoid arthritis is a chronic inflammatory disorder that mainly affects the joints (Lee/Weinblatt, 2001). It is the most common form of arthritis, affecting about 1 percent of the total population in a female/male ratio of 2,5. The disease can occur at any age, but is most common for people between 40 and 70, with the likelihood of occurrence rising as function of age. In terms of cost, disability and lost productivity, rheumatoid arthritis has a substantial societal effect.

Until today, the cause of rheumatoid arthritis is incompletely understood. The inflammation of the joints is a chronic tissue-specific, which is affected by several immune responses. The disease involves a complex interplay among genotype, environmental triggers and chance (McInnes/Schett, 2011).

Today, the disease is incurable, but an assortment of drugs (Disease Modifying Antirheumatic Drugs = DMARDs) is available to retard its progression (Smolen et al., 2010). The treatment and administration of these drugs must be monitored in a continually manner and adjusted according to the patient's situation. Furthermore, it needs to be emphasized that such a therapy is usually administered over the long term, stretching over several decades. As this long-term therapy creates big amounts of data, the monitoring process is supported by RCQM.

3.1.2. RCQM

The RCQM system was developed in an joint effort of the Institute for Medical Informatics, Statistics and Documentation and the Division of Rheumatology and Immunology at the Medical University of Graz, with the aim to speed up the heavily time-constrained process of collecting the clinical and functional parameters of patients with rheumatoid arthritis, calculating the medical score values and displaying the data as function of time. The parameters are collected in a standardized form using an interactive graphical representation of the human body, as shown in Figure 26.

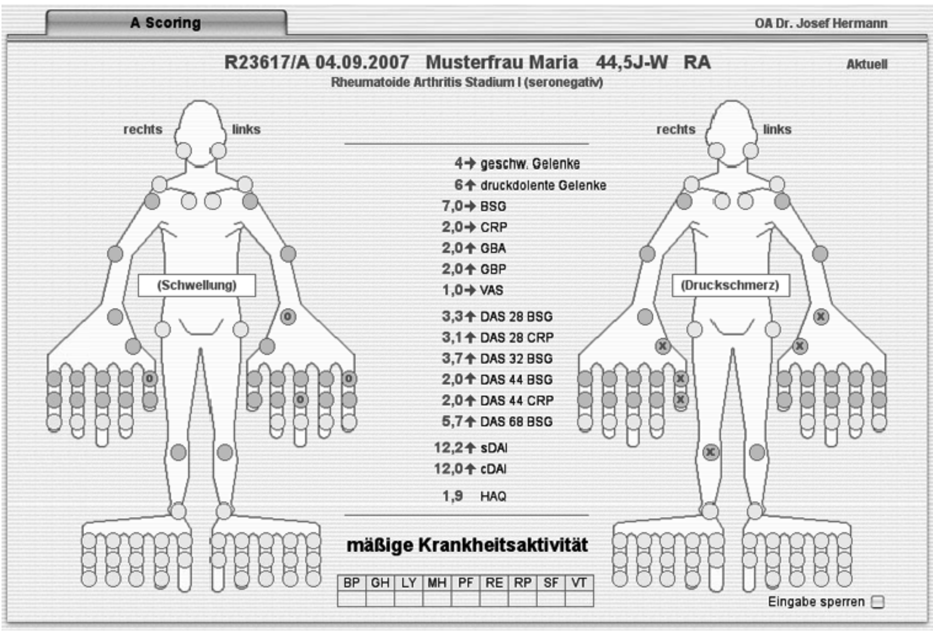


Figure 26: RCQM clinical parameter collection mask

The interface contains all 72 joints that are examined for swelling and tenderness. After the data collection phase is finished, the system automatically calculates a number of scores (including the DAS score). Together with previously collected data tuples, these scores form the information base for the analysis of the current disease activity.

Another important feature of RCQM is the fact that it provides decision support with regard to the optimization of the treatment strategy. According to Simonic et al. (2011) the main medical goals for the treatment are as follows:

- Prevention of disease progression and functional deterioration
- Absence of swollen joints
- Normalization of inflammation parameters
- Freedom from pain and/or complaints

These goals were translated into the following standard operating procedures of RCQM (Simonic et al., 2011):

- High disease activity (DAS28 ≥ 5.1): This represents acute danger of damage to the joints. The treatment goal is to reduce disease activity to moderate values as quickly as possible with a high dosage of DMARDs and supplemental biologicals. Patients should be monitored every 1-2 weeks.
- Moderate disease activity: (3.2 ≤ DAS28 < 5.1): The aim is to reduce the score with an individually optimized/balanced mono therapy or combined therapy with different DMARDs and optional biologicals, with respect to the course of disease, swelling, pain, adverse effects, and tolerance. Patients should be monitored at intervals of 1-3 months.
- Low disease activity (DAS28 < 3.2): This represents the endpoint of the disease activity minimization procedure. The aim here is to retain the status with an optimized/minimized administration of medications. Check-ups should be scheduled every 3 months.

As the goal is to keep the medication as low as possible, this strategy typically leads to values near the upper limit of the low bound (DAS28 < 3.2) over time. Figure 27 shows the reporting screen of RCQM. The temporal course of the disease (middle of the screen) is combined with information regarding the medical treatment (bottom of the screen). Additionally, the user can select, which clinical reporting parameters to include into the time series representation (right side of the screenshot).

From an operational perspective, the support of RCQM shows, among others, the following advantages for a rheumatologist (Simonic et al., 2011):

- Realization of division-of-labor processes without loss of quality

- Retrospective evaluation of the administered treatment (effectiveness check)
- Differential-diagnostic evaluation of the current situation
- Individual optimization of the use of medications in the context of the defined treatment goals

Additionally, RCQM provides the function to generate semi-automatic medical reports and also assists the communication process between physician and patient.

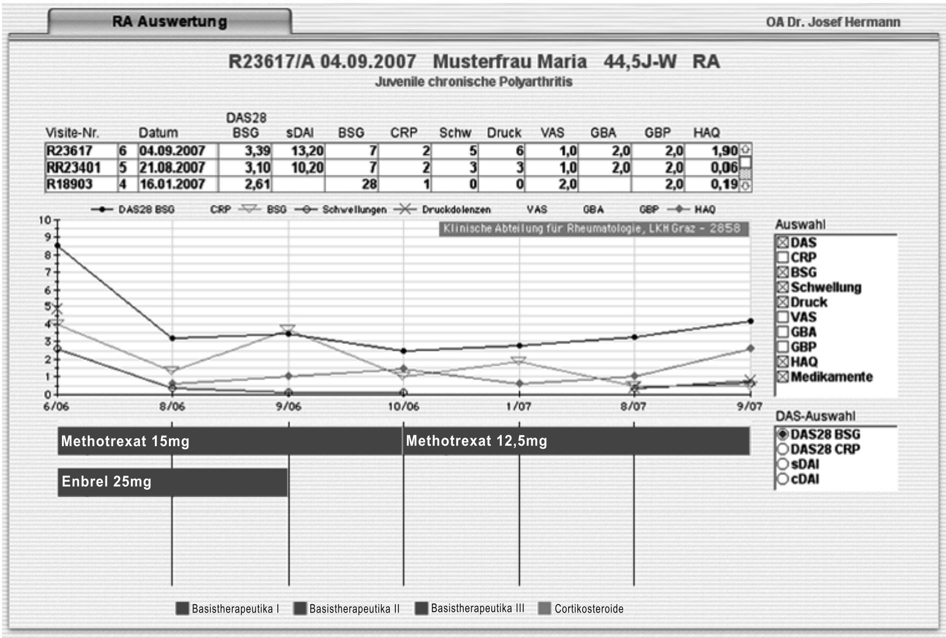


Figure 27 Reporting screen of RCQM

3.2. Other Medical Information Systems

LifeLines (Plaisant et al., 1996) is a Java desktop application for visualizing electronic patient histories. Problems, diagnoses, test results and medications are displayed as dots or horizontal lines. The interface allows zooming out to get an overview of the overall patient history and zooming in to examine a region of interest in more detail. Line and dot strength and color represent relationships and significance.

TimeLine (Bui et al., 2007) provides problem-centric temporal visualization for medical data. The system reorganizes the huge amount of electronic medical patient data around medical disease histories and conditions. For this reorganization step, the authors provide a formal set of mechanisms for mapping and data transformation. Furthermore, TimeLine offers a conceptual abstraction for connecting patient data and visual representation for several diseases.

LifeLines2 (Wang et al. 2010) is an electronic health record system, that – as the authors emphasize – does not only concentrate on the tasks of storage, retrieval and exchange of patient data, but also cares about the end user. End users for this application are primarily medical personnel. After collaborating research with physicians over two and a half years, the authors put the main focus on temporal categorical data analysis tasks. LifeLines2 was deployed in the context of eight case studies in hospitals. The authors generalized the feedback and application usage logs into several recommendations for future electronic health record applications:

- *Use Alignments* – when an alignment becomes active, the timeline should scale in relation to the active alignment.
- *Show details* – a detailed depiction of patient records is important. Even if the primary view of the data was supposed to be in an overview.
- *Support richer exploration process* – information systems need to support branched search, history keeping and backtracking.
- *Flexible data types* – visualization systems should support various data types (e.g. numerical values and categories).

- *Higher information density* – the amount of data a physician wants to see is typically much larger than a screen can hold. Therefore, a lot of scrolling is involved. To reduce scrolling, it is important to find the right amount of information density for the screen.

Klimov et al. (2010) introduce the VISualizatIon of Time-Oriented RecordS (VISITORS) system for display and analysis of multiple patient records at the same time. This is especially helpful for the visual inspection of clinical trial results and quality assessment purposes. Additionally, the software allows to aggregate data from different patients by defining aggregation rules. This aggregation enables physicians to identify patient groups with similar disease profiles.

Chittaro (2006) developed a mobile application for accessing, analyzing and updating patients' medical records. The paper discusses the problems and challenges connected with the visualization of patient data on small-screen devices. The device used for this research was a Windows PocketPC. Two different interfaces were developed: one with different aspects displayed on different parts of the screen and one with an overlapping display of different data types. Both interfaces allow the fade-in/out of different data types. Figure 28 shows the two implemented interface types. As the author implemented only one informal user interface test with one physician, there is no available data about the applicability of the user interface designs in a real-world environment.

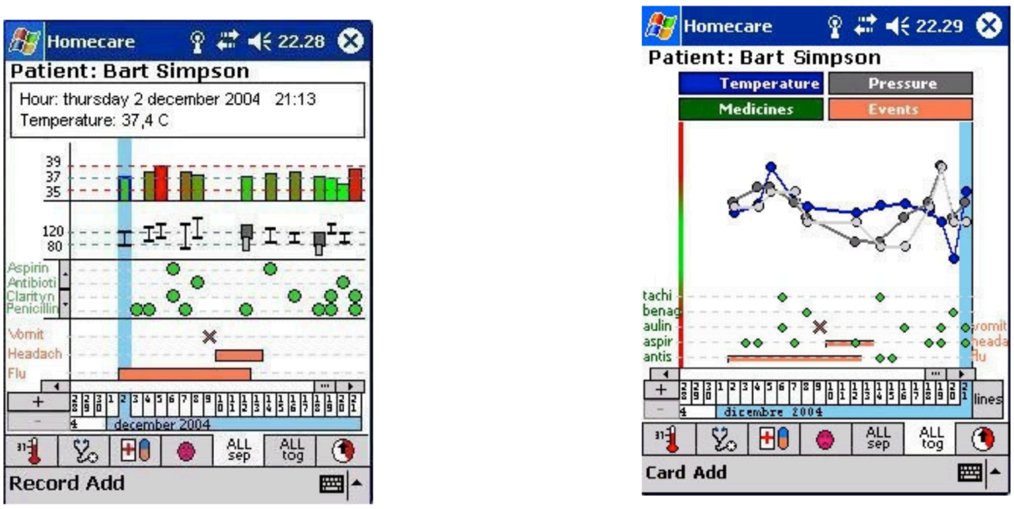


Figure 28: Two different interface prototypes for Windows PocketPC (Chittaro, 2006)

Ardito et al. (2006) introduced PHiP (Patient History in Pocket), a tool designed to display and query the histories of patients affected by epilepsy on mobile devices (PDA). The data was taken from a hospital database (Giovanni XXIII pediatric hospital in Bari, Italy). The authors' main challenge was to display as much patient data on the limited screen area as possible. For this task they implemented two different interface designs. First, an overview and detail interface and second, a zoomable interface. On each development stage, informal user interface tests were conducted with two to five neurologists. They found that zooming and panning are very efficient operations that provide neurologists a better overview of the patient history, without restricting the possibility of focusing on detailed views. Moreover, varying the displayed time periods is faster on the zoomable interface than moving the lens on the overview screen and getting the corresponding information on the focus screen. Furthermore, the authors suggest that although the application was designed for patients affected by epilepsy, it can also be used for various other diseases.

4. Implementation

This chapter provides an overview of the implementation of the RCQMmobile application.
The first section describes some of the technologies used to build apps. After this, a rough overview of the application structure is presented.

4.1. Technology

For the implementation of RCQMmobile, a number of hardware and software tools have been used. In the following subsections, the most important concepts are introduced.

4.1.1. iPad Device

The iPad was chosen as target device category for developing RCQMmobile. The iPad is a popular tablet device developed by Apple Inc. and built by the Taiwanese company Foxconn (Mashman, 2011). Its recent version, the iPad Air, has a 9.7-inch LED-backlit multi-touch display with IPS technology, which supports a resolution of 2048-by-1536 (264 pixels per inch). It is powered by an Apple A7 chip, a 64-Bit-ARM-CPU. The processor contains two 1.3 GHz RISC cores which support the 64-bit ARMv8 instruction set.
One main reason for choosing the iPad over similar Android devices was the higher market share of iPads within the enterprise sector. According to Apple's Q1/14 conference call, the iPad made up for 90 percent of all corporate tablet activations in that quarter and accounted for 78 percent of the total US enterprise tablet market[3]. From a development point of view, both platforms offer roughly the same possibilities and furthermore development effort does not vary substantially between the two platforms.

[3] For a full transcript of the call, compare: http://seekingalpha.com/article/1971291-apples-ceo-discusses-f1q-2014-results-earnings-call-transcript?page=1

4.1.2. iOS Operating System

iOS (formerly iPhone OS) is an operating system for mobile devices designed and marketed by Apple Inc. It is derived from MacOS X, with which it shares several application frameworks (like Foundation) and the Unix Darwin Core.

The current version is iOS7, which was released on September 18, 2013. Up to now, there are one-year release cycles for major OS upgrades.

The numerous APIs of iOS7 provide a lot of functionality needed for implementing RCQMmobile. The following paragraph gives an overview of the “out-of-the-box” capabilities of iOS.

iOS consist of four abstraction layers: Core OS layer, Core Services layer, Media layer and Cocoa Touch layer (Figure 29).

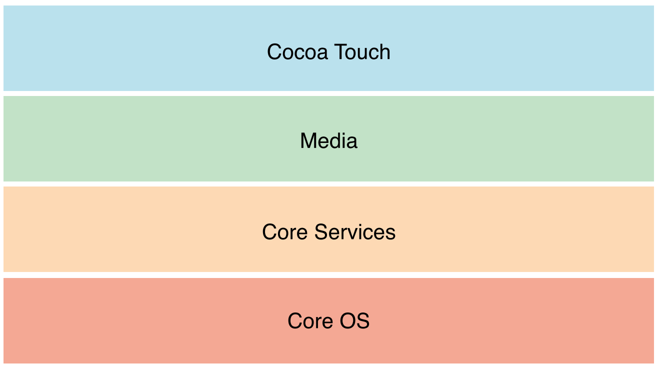


Figure 29 Layers of the iOS system

Apple delivers the Cocoa Touch API, a large set of frameworks, to access functions in all of the mentioned layers. The following frameworks within Cocoa Touch were used for building the application:

- *UIKit*

  A huge framework that provides classes for constructing and managing the user interface of the application. Inter alia, it provides the application object, event handling, drawing model, windows, views and controls for interacting with a touch screen device.

- *Foundation*

  Provides the base layer of Objective-C classes. In addition to defining primitive object classes, it introduces several basic utility classes as well as classes to support programming paradigms that are not covered by the Objective-C language. Furthermore, the framework contains the root object class NSObject, from which all other framework classes on the iOS platform derive.
- *QuartzCore*

  Provides animation classes and image/video processing classes.
- *CoreGraphics*

  Based on *QuartzCore*, this framework provides advanced drawing classes.

Beside this, a few open-source third-party libraries were used. The following chapters give a short introduction of these frameworks.

4.1.3. Plotting: CorePlot Framework

CorePlot (CorePlot, 2013) is an extensive open-source plotting framework, which works on both iOS and MacOS platforms. It is build upon Apple frameworks like Quartz Core and Core Animation and is distributed with permissive BSD licenses. Furthermore, the framework is well-tested and implements ARC (automatic reference counting), which is a great benefit in terms of stability and error avoidance.
The utilization of Core Animation implies that many elements of the graph can be animated with transition effects. These animations can lead to additional value concerning UX, by allowing the generation of better visual feedback to users' interactions.

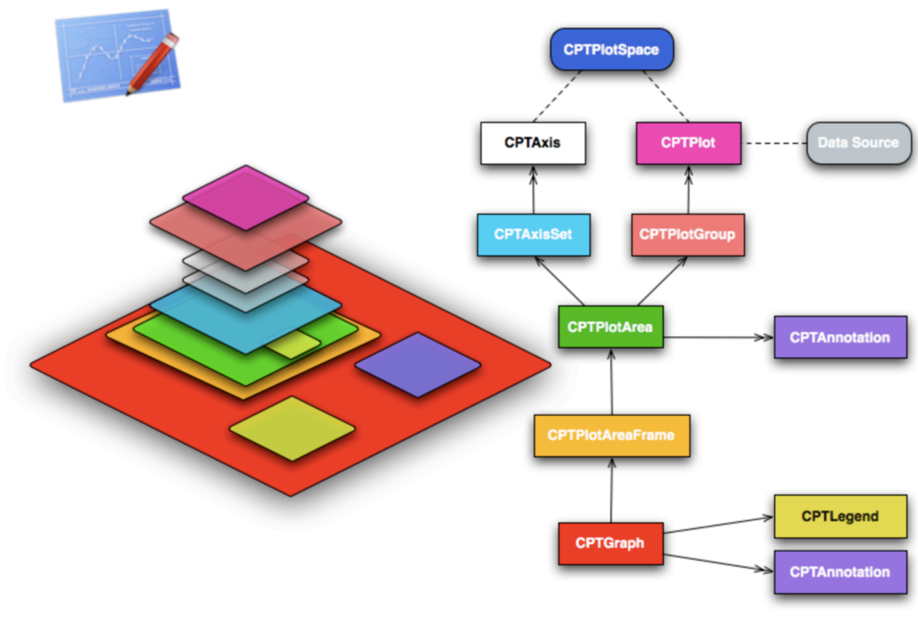


Figure 30 - Relationship diagram of the core plot framework

Figure 30 shows a class-relationship diagram of the CorePlot framework. The most important classes within the CorePlot framework are CPTGraph and CPTPlot.

CPTGraph

The central, top-level class within the CorePlot framework is CPTGraph. The term "graph" comprises in the corePlot semantic the whole diagram, including plots, title, labels, legends, annotations and axes. CPTGraph is formulated as an abstract class, from which all different supported graph classes derive. Basically, a graph is a factory class, which is responsible for constructing all subcomponents that make up the graphic and for setting up appropriate relationships. For instance, concrete subclasses from CPTGraph are CPTXYGraph.

CPTPlot

A plot is a particular representation of data within the graph. Again, CPTPlot is an abstract class, from which concrete classes for different plot types derive. By default, CorePlot supports the following plot types:

- CPTBarPlot

- CPTPieChart
- CPTRangePlot
- CPTScatterPlot
- CPTTradingRangePlot

Thereby, a range plot is a graph, in which each data point can contain a value range and is visualized by indicating this range. Related to this plot type is the trading range plot. In this case, each data point has a range, which extends over minimum, maximum, open and close values. The scatter plot is an ordinary x,y-chart with a one-to-one relationship between x and y values.

4.1.1. Networking Capabilities: AFNetworking 2.0

AFNetworking (available for free on github.com) is an extensive networking library designed for the Mac OS and iOS platforms. It is build on top of the Foundation Framework's URL Loading System. The development of the open-source library is community-driven with around 180 contributors and 3000 forks. Today, it is the most widely used open-source library within the iOS community (more than 10,000 apps built with it).

4.1.1. Sourcecontrol: Bitbucket

Bitbucket is a web-based hosting service for projects using Git revision control systems. The platform offers both free accounts and commercial plans. For RCQMmobile, a free account was installed.

4.2. Converting the RCQM interface from desktop to mobile/touch environment

The UI of RCQMmobile was designed with two key principles in mind.
First, it was considered important to take full advantage of the multi-touch device.
Thereby, it was a requirement to use gestures which are familiar to users of other multi-touch applications. The usage of gesture commands in such a consistent way is

especially important as gesture commands are by nature invisible to the user – in contrast to e.g. a command button. Undiscovered gestures can therefore limit the perceived feature set of the application and subsequently influence the user experience negatively. Additionally, as tablet devices – in contrast to the desktop devices – can be rotated, we had to deal with portrait and landscape layouts. We decided to use the rotation feature to emphasize the list view in the portrait mode whereas we featured the chart-view in the landscape mode in more detail.

Second, it was also required to implement the same scope of functions as in the desktop application. Therefore, for porting the interface from desktop to touch, we analyzed the different information areas of the desktop application in a first step. Within this step, also other chart interfaces, like stock market data interfaces, were analyzed to discover similarities with RCQM. This additional step enabled us to design the RCQM interface in a generic way to also support other types of longitudinal data with minimal additional adaption effort. Figure 31 and Figure 32 show two different desktop interfaces with longitudinal data visualizations. On both visualizations, the identified information areas are color-coded.

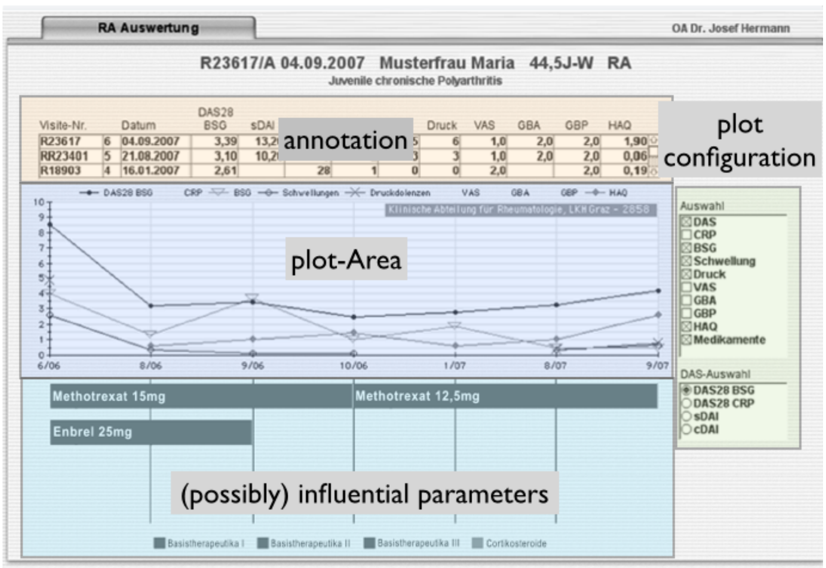


Figure 31 – Areas of Information - RCQM desktop application

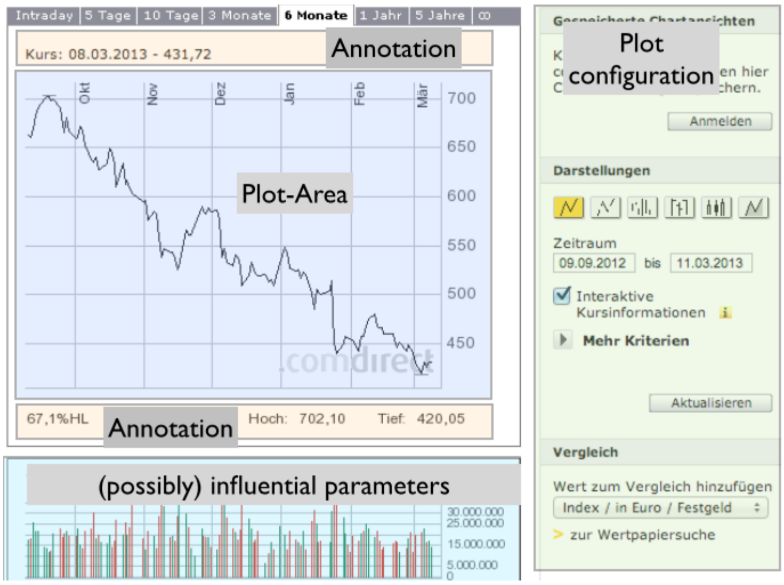
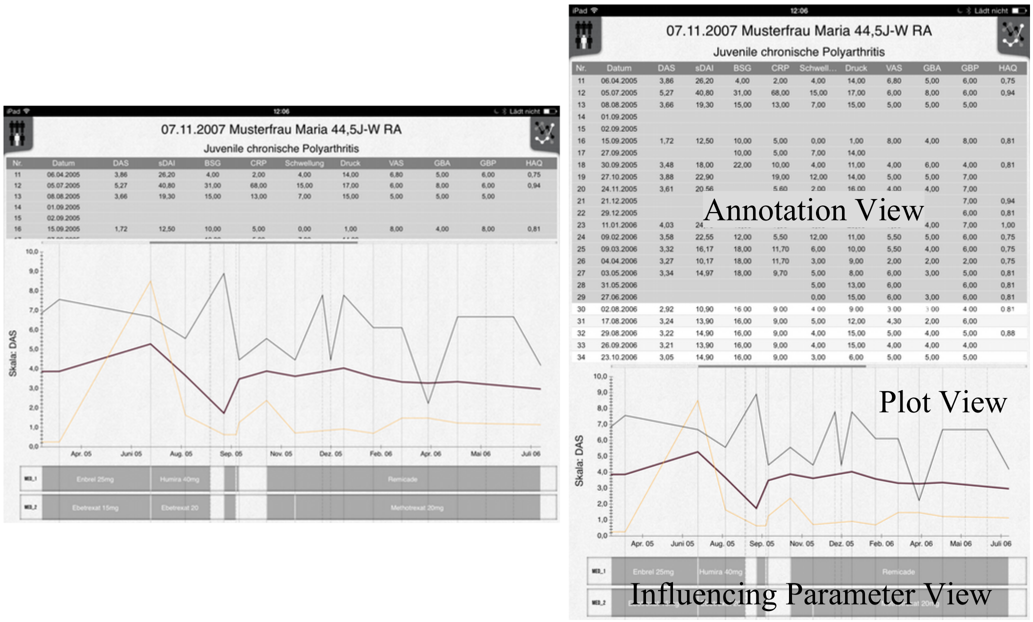


Figure 32 – Areas of Information - www.comdirect.de, a stock price visualization

As shown in the figures, both interfaces comprise roughly the same types of information areas. Differences exist in the kind of visualization of concrete information areas. For



instance, whereas drugs are the influential parameter in the RCQM case, the trading volume is displayed in the stock market case. Both types of data are displayed by using different kinds of concrete data visualization. The annotation field differs in such a way that RCQM shows the whole raw data in a table, whereas it is displayed inline in the stock market case.

Figure 33 - Landscape and portrait view of RCQMmobile (menu closed)

Figure 33 shows the main interface design of the final RCQMmobile version in portrait and landscape mode. Both menus for selecting patients and chart configuration are closed in this figure. As can be seen, in contrast to the desktop version of RCQM (Figure 31), the chart configuration area does not consume any screen estate by default. Instead it can easily be displayed on-demand by either panning from the right screen edge or tapping the button on the top right side.

The aspect ratio of the plot area was chosen in a way to maximize the visibility of changes. This ratio was taken for both portrait and landscape layout. Furthermore, for optimal use of the screen estate, the scaling of the y-axis adapts automatically to the chosen reference plotline. All other plotlines are scaled to this reference size.

Before opening the plot screen, RCQMmobile displays a login dialog (Figure 34). The user needs to authorize via username and password and receives a token for querying patient information data from the web service in return.

Figure 34 - Login Screen of RCQM

4.3. Implemented Gestures

In comparison to a traditional mouse-based device, a great deal of commando input on a multi-touch appliance is gesture-based. Gestures can provide a quick and easy way to access common functions of an application. Furthermore, the implementation of a gesture-based interface allows reducing the number of visible commando elements on the screen. This is especially important as a mobile tablet device usually comes with very limited screen size. Hence, several gestures were implemented to enrich the interaction possibilities with the displayed longitudinal data.

For implementing gesture recognition, Apple provides the abstract UIGestureRecognizer (UIGestureRecognizer Class Reference, 2013) class within its UIKit framework for subclassing. A gesture-recognizer object decouples the logic for recognizing a gesture and the generated action on that recognition. This decoupling mechanism is provided by an implementation of the command design pattern, which in Cocoa is called Target/Action[4]. When a gesture recognizer, which can be applied to a view frame, recognizes a gesture, it sends an action message to each designated target object.

Gesture recognizers come in two types: continuous and discrete. Discrete gesture recognizers send a single action after a full gesture is recognized. Continuous gesture recognizers send a series of action messages during the gesture is executed.

Within UIKit, Apple provides some out-of-the-box UIGestureRecognizer subclasses for common gestures:

- *UITabGestureRecognizer* (discrete) – recognizes single or multiple taps with a specified number of fingers.
- *UIPinchGestureRecognizer* (continuous) – recognizes two-finger pinching. When the user moves the two fingers towards each other, the conventional

[4] Strictly speaking, the command pattern is not fully implemented in the Target/Action mechanism of Cocoa as the request is not encapsulated into a separate object. Nevertheless, it offers roughly the same flexibility as the command pattern (Napier/Kumar, 2013).

meaning is zoom-out. Moving the fingers away from each other, means zoom-in.

- *UIRotationGestureRecognizer (*continuous*)* – recognizes two-finger rotation. When the user moves tow fingers in a circular motion at the same, a rotation is detected.
- *UISwipeGestureRecognizer* (discrete) – recognizes swipe gestures in one or more directions. A gesture is considered a swipe, when the specified minimum number of fingers has moved far enough.
- *UIPanGestureRecognizer (*continuous*)* – recognizes panning (dragging) gestures with one or more fingers. The user must tap and hold one or more fingers on the view and pan it to trigger an action. The action begins when the specified minimum number of fingers has moved far enough to be considered a pan.
- *UIScreenEdgePanGestureRecognizer* (continuous) – subclass of UIPanGesture-Recognizer. Detects panning gestures that start near an edge of the screen.
- *UILongPressGestureRecognizer* (continuous) – recognizes long-press gestures. A user must tap and hold one or more fingers on a view and hold them for a minimum amount of time before the action is triggered. While tap and hold, the fingers must not move more than a specified distance.

In addition to these out-of-the-box gesture recognizers, it is possible to implement a custom subclass of UIGestureRecognizer to support new gestures.

For RCQMmobile a UICutGestureRecognizer was implemented to allow the user to apply a “cut-gesture” on the chart. This gesture enables the user to quickly zoom into a part of the graph that is defined by the cut. A cut is defined as a parallel movement of two fingers in one direction. Thereby, the two fingers must have a minimum distance of at least one fingertip and the movement must fulfill a minimum horizontal distance of at least four fingertips. Figure 35 shows a visualization of the cut gesture.

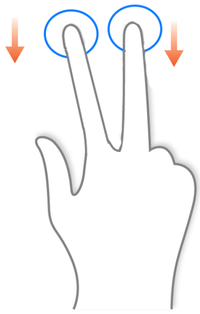


Figure 35 - Cut Gesture

In total, the following gesture commands have been implemented for RCQMmobile:

Graph: Zoom in/out	Pinch/Spread gesture
Graph: Quick zoom-in to defined area	Cut gesture
Graph: Quick zoom-out to full view	Double-Tap Gesture
Graph: Moving forward / backwards	Pan Gesture
Graph: Showing visit detail	Long Press gesture
Menu: Show graph options	Screen-edge pan from the right side; tap and hold on top-right-corner button.
Menu: Show patient selection	Screen edge pan from the left side; tap and hold on top-left-corner button.

Table 1 – Implemented gestures in RCQMmobile

4.4. Synchronization of information areas

When zooming into the chart area to view some subpart of the chart in more detail, the application keeps the drug area as well as the annotation area in sync with the plot view. Within the annotation area, all currently visible visits are highlighted in a dark gray color.

When a user executes a long-press on one specific visit, it is highlighted in a red color within the annotation view.

4.5. Data Exchange Model

In this chapter the data exchange format for the communication between data server and mobile device is introduced. As the mobile device is used in the stated scenario primarily for data consumption and not for the creation of new data, communication takes place in form of secure https-GET requests from the mobile device to the data server. For our use case, we require two different kinds of requests. For illustration purposes, JSON notation is used for the sample queries. The same functionality can be implemented in a straightforward way using other transport formats like XML.
Although the example data reflects a special kind of disease, it should be noted that the introduced exchange format could be applied to wide range of use cases, specifically to all tasks of measuring the impact of any kind of on/off variable on score values over time.

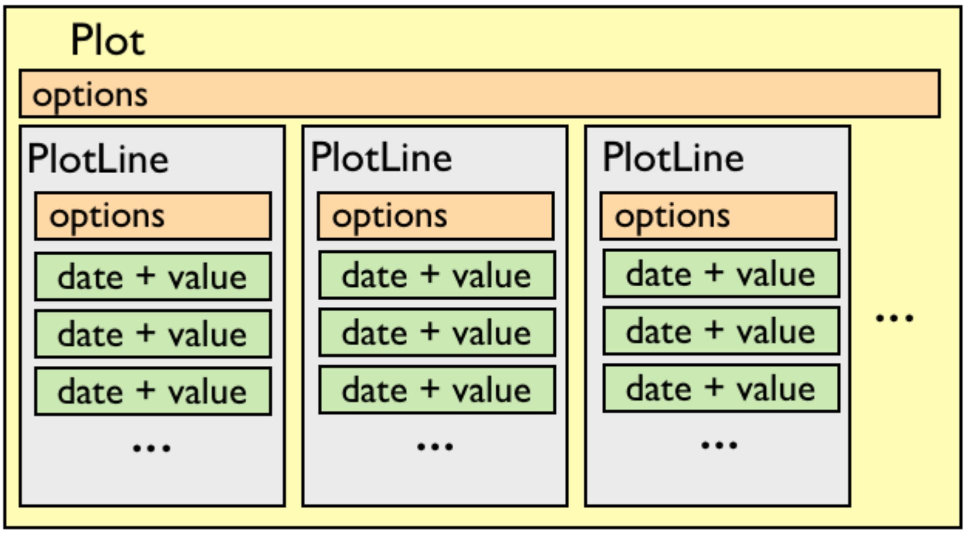


Figure 36 – Overview of the data exchange format

List of patients query

This query is the starting point for the application. The application triggers the request for the URI *[baseurl]/patientlist?access_token={access_token}* and retrieves a list of patients affiliated with the UID of the app user. This list contains the basic patient information like name and age, to give an application user enough information to select a patient for detailed inspection.

```
{
 "PatientList" : [
   {
    "PatientID" : "42",
    "FirstName" : "Max",
    "LastName" : "Mustermann",
    "Age" : 60
   },
   {
    "PatientID" : "43",
    "FirstName" : "Margarethe",
    "LastName" : "Musterfrau",
    "Age" : 55
   }
 ]
}
```

Query for longitudinal patient data

This query type can be used in two ways. First, by calling *[baseurl]/patients/all*, the application retrieves an array of longitudinal patient data for all patients affiliated with the UID of the app user triggering the request, or second, by calling [baseurl]/patients/[PatientID]?access_token={access_token} data for one specific patient. Either way, one patient element within the response is constructed in the same way for both requests.

On the top layer, one patient response encapsulation consists of *ID*, *name*, *age*, *gender* and *patientData,* which is an encapsulation object for the longitudinal patient data. *patientData* is structured into a *MetaData* object and into a *VisitsData* object. *MetaData* contains additional general information regarding the specific score and drug data used within *VisitsData*.

The array *VisitsData* contains one *Visit* object for each medical visit of a patient. Every *Visit* object includes general visit information like *date*, *visitID*, attending *physician* and the *disease* and, furthermore, measurement *score data* and the administered *drugs*.

```
{
 "Patients" : [
   {
    "PatientID" : 42,
```

"LastName" : "Musterman",
"FirstName" : "Max",
"Age" : 60,
"Gender" : "m",
"PatientData" : {
  "MetaData" : {
    "Drugs" : {
      "Methotrexat 15mg" : {
        "Class" : "Basistherapeutika 1"
      }
    },
    "Scores" : {
      "CRP" : {
        "LimitValue" : 12.5
      }
    }
  },
  "VisitsData" : [
    {
      "Disease" : "Juvenile chronische Polyarthritis",
      "Physician" : "Dr. Musterfrau",
      "Date" : "2012-04-25T19:19Z",
      "VisitID" : "10343",
      "Data" : {
        "ScoreData" : {
          "DAS" : {
            "DAS28 BSG" : 5,
            "DAS28 CRP" : 4.3,
            "sDAI" : 3.3,
            "cDAI" : 5.8
          },
          "CRP" : 4,
          "BSG" : 1.2,
          "Schwellung" : 3.2,
          "Druck" : 4.2,
          "VAS" : 1.1,
          "GBA" : 2.1,
          "HAQ" : 0.1
        },
        "DrugData" : {
          "Methotrexat 15mg" : true,
          "Enbrel 25mg" : true
        }
      }
    }
  ]
}

```
    }
  ]
}
```


4.5.1. HTML vs. native implementation

Nowadays, with a relatively strong fragmentation into different mobile computing platforms (e.g. Android, iOS, WindowsMobile, Blackberry OS) it is appropriate to think about developing cross platforms by utilizing html and javascript technologies.
At least from an economic point of view, the step from native to web code clearly makes sense. Instead of writing a new code base for each supported platform (and furthermore taking the effort of maintaining each of them), web code allows to basically create one application, which supports all different platforms.
Nevertheless, from a UX and performance viewpoint (for a very detailed performance comparison see Crawford (2013)), web implementations still have deficits compared to their native counterparts (Charland/Leroux, 2011).
First, each mobile platform has slightly different UI guidelines and UI elements. This implies either additional customization effort for supporting all different layouts or a compromised UX.
Second, a web implementation is more restricted concerning the integration of platform specific features, like accelerometer data or operating system notifications.
Further, from a performance perspective, Java Script is much slower than a native implementation. This fact is especially important as the resources on a mobile device are very limited compared to desktop devices. According to Crawford (2013), Java Script on a mobile platform is about 5 times slower than native code.

4.5.2. Testing

Testdata

Anonymized patient histories have been used for testing data visualization in RCQMmobile. The sample consisted of data from 175 patients with a total of 1782 visits (minimum: 6, maximum: 27, median: 11 visits).

The data was extracted from the RCQM desktop system in .csv format and was afterwards converted into the defined JSON transportation format, which was introduced earlier in chapter 4.5. For the automatic data conversion, a python script was implemented.

Unit tests: SenTestingKit

The model logic of the RCQMmobile was tested using SenTestingKit. The SenTestingKit is Apple's install of OCUnit. As Xcode4 includes the SenTestingKit (out-of-the-box), there is almost no integration effort necessary to add OCUnit-Tests. Therefore, it was inviting to use the OCUnit-Tests for the implementation of unit tests.
Concerning the test construction, a similar approach as described by Lee (2013) was pursued. Thereby, tests for the setter and getter logic, as well as data preparation and obtaining were implemented. The network connection was tested by using a mock object. This object implements the defined network interface and utilizes the already mentioned test dataset, which is saved locally in a file.

Prototype Distribution: Testflight

For managing prototype distribution and crash reporting of RCQMmobile the Testflight platform was used (freely available at http://testflightapp.com). Testflight is an over-the-air iOS beta distribution platform. Furthermore, Testflight allows the developer to collect an array of usage statistics of application users. The tracking is easily integrated by including the open-source Testflight framework into the app

5. Conclusion

The aim of this thesis was to give an overview of various methods for longitudinal data visualization and multi-touch interaction and introduce RCQMmobile – a mobile multi-touch development of the RCQM system.

Several key findings were established during the research process.

On longitudinal data visualization

Our findings show that it is of uppermost importance to understand longitudinal data visualization as a tool to convey the underlying message of raw data (and not as a mean by itself). As stated by Tufte et al. (1993), data visualizations are instruments for reasoning about quantitative information at their best.

Thereby, it is important to think about several aspects of the visualization:

- Choose the type of visualization, which fits the data best (sequence charts, bar charts, line charts, circle charts)
- Avoid "chart-junk" – maximize the data/ink ratio
- Highlight missing values
- Choose the right aspect ratio
- Display the data in the right granularity, with possibility to zoom in/out

On multi-touch interaction

First, there are some well-identified benefits of multi-touch input compared to mouse-based input. To state some of them:

- Faster direct selection of points
- Higher degree of freedom in interaction
- No additional peripherals needed for input
- Reduced number of UI command elements
- Complex mouse operations like select and rotate

Second, for high precision tasks, indirect selection via mouse cursor and point device outperforms direct-touch input.

Third, as multi-touch is a relatively new technology, there are no well-defined gestural interaction standards. Furthermore, many information visualizations are not well adapted to this new interaction method but rather emulate the handling of a mouse-based interface.
Last, when creating applications for multi-touch devices (especially when porting existing applications from mouse-based devices to multi-touch) it is very important to create a new UI from scratch, which is specifically tailored to the device properties. Thereby, not the device specific interaction properties are used, but also the interface design to the UI-specific properties of the platform are adapted.

On the combination of longitudinal data visualization and multi-touch interaction

Our combined findings point into the direction that the combination of longitudinal data visualization and multi-touch interaction can be overall beneficiary.
The benefit arises because many interactions with (especially large) longitudinal data sets are connected with zooming into a region and scrolling around almost simultaneously. These complex, coupled interactions are more easily achieved with multi-touch devices than with mouse-based controls.
In RCQMmobile, we incorporated these findings by implementing a set of gestures for graph interaction. In total, we implemented seven gestures for efficient data interaction. Within this gesture set, a new two-finger “cut” gesture for quickly zooming into portions of the chart is available.

List of Figures

6. List of Tables

7. References

Ahlberg, Christopher, Ben Shneiderman. “Visual information seeking: tight coupling of dynamic query filters with starfield displays.” Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1994.

Amarasingham, R., Plantinga, L., Diener-West, M., Gaskin, D. J., Powe, N. R. (2009). Clinical information technologies and inpatient outcomes: a multiple hospital study. Archives of Internal Medicine, 169(2), 108.Buntin, Melinda Beeuwkes, et al. "The Benefits Of Health Information Technology: A Review." Health Affairs 30.3 (2011): 464-471.

Archambault, D., Purchase, H., Pinaud, B. (2011). Animation, small multiples, and the effect of mental map preservation in dynamic graphs. Visualization and Computer Graphics, IEEE Transactions on, 17(4), 539-552.

Ardito, C., Buono, P., Costabile, M. F., Lanzilotti, R. (2006). Two Different Interfaces to Visualize Patient Histories on a PDA.

Aris, A., Shneiderman, B., Plaisant, C., Shmueli, G., Jank, W. (2005). Representing unevenly-spaced time series data for visualization and interactive exploration. In Human-Computer Interaction-INTERACT 2005 (pp. 835-846). Springer Berlin Heidelberg.

Bisgaard, S., Kulahci, M. (2011). Time series analysis and forecasting by example. John Wiley & Sons.

Box, G.E.P and G.M. Jenkins, Time Series Analysis, Forecasting, and Control, Holden-Day, Oakland, CA, 1970.

Bui, A. A., Aberle, D. R., Kangarloo, H. (2007). TimeLine: Visualizing integrated patient records. Information Technology in Biomedicine, IEEE Transactions on, 11(4), 462-473.

Buntin, M. B., Burke, M. F., Hoaglin, M. C., Blumenthal, D. (2011). The benefits of health information technology: a review of the recent literature shows predominantly positive results. Health Affairs, 30(3), 464-471.

Buxton, B. (2007). Multi-touch systems that I have known and loved. Microsoft Research, 1-10.

Brockwell, P. J., Davis, R. A. (2009). Time series: theory and methods. Springer.

CorePlot (2013). Core Plot Design Overview. https://code.google.com/p/core-plot/wiki/HighLevelDesignOverview. Last access 10/2013.

Charland, A., Leroux, B. (2011). Mobile application development: web vs. native. Communications of the ACM, 54(5), 49-53.

Chatfield, C. (2003). The analysis of time series: an introduction. CRC press.

Chittaro, L. (2006). Visualization of patient data at different temporal granularities on mobile devices. In Proceedings of the working conference on Advanced visual interfaces (pp. 484-487). ACM.

Crawford D. (2013). Why mobile web apps are slow. Available under: http://sealedabstract.com/rants/why-mobile-web-apps-are-slow/ (02/2014)

Fleisher, R. B. Bullinger, T. A. (1998) Star Plots of analyte values over time. EP0877249. *G01N33/48.* ORTHO CLINICAL DIAGNOSTICS INC. Christopher Paul Mercer, Carpmaels & Ransford 43, Bloomsbury Square London WC1A2RA

Frisch, M., Heydekorn, J., Dachselt, R. (2009). Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (pp. 149-156). ACM.

Gonzalez, H., Halevy, A., Jensen, C. S., Langen, A., Madhavan, J., Shapley, R., Shen, W. (2010, June). Google fusion tables: data management, integration and collaboration in the cloud. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 175-180). ACM.

Harris, R. L. (1999). Information Graphics: A Comprehensive Illustrated Reference: [visual Tools for Analyzing, Managing, and Communicating]. Oxford University Press.

Holzinger, A. (2002) *Multimedia Basics, Volume 3: Design. Developmental Fundamentals of multimedial Information Systems.* New Delhi, Laxmi Publications.

Holzinger, A. (2005) Usability Engineering for Software Developers. *Communications of the ACM,* 48, 1, 71-74.

Holzinger, A., Stickel, C., Fassold, M., Ebner, M. (2009) Seeing the System through the End Users' Eyes: Shadow Expert Technique for Evaluating the Consistency of a Learning Management System. In: Holzinger, A., Miesenberger, K. (Eds.) *HCI* *and Usability for e-Inclusion. 5th Symposium of the Austrian Computer Society, USAB 2009, Lecture Notes in Computer Science LNCS 5889*. Heidelberg, Berlin, New York, Springer, 178-192.

iPad Tech Spec (2013). iPad Air Technical Specifications. http://www.apple.com/at/ipad-air/specs/. Last access 10/2013.

Isenberg, P., Isenberg, T. (2013). Visualization on Interactive Surfaces: A Research Overview. i-com.

Kin, K., Agrawala, M., DeRose, T. (2009). Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In Proceedings of Graphics interface 2009 (pp. 119-124). Canadian Information Processing Society.

Klimov, D., Shahar, Y., Taieb-Maimon, M. (2010). Intelligent visualization and exploration of time-oriented data of multiple patients. Artificial intelligence in medicine, 49(1), 11-31.

Lee D. M., Weinblatt M. E., “Rheumatoid arthritis,” Lancet, vol. 358, no. 9285, pp. 903–11, Sep 2001. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0140673601060755

Lee, B., Isenberg, P., Riche, N. H., Carpendale, S. (2012). Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 18(12).

Lee, G. (2012). Test-driven IOS Development. Addison-Wesley.

Muller, L. Y. (2008). Multi-touch displays: design, applications and performance evaluation. Grid Computing–Master’s Thesis.

Musen, M. A., Helder, J. C. (1997). Handbook of Medical Informatics.

Hao, M. C., Dayal, U., Keim, D. A., Schreck, T. (2007, May). Multi-resolution techniques for visual exploration of large time series data. In EuroVis (pp. 27-34).

Jerri, A. J. (1977). The Shannon sampling theorem—Its various extensions and applications: A tutorial review. Proceedings of the IEEE, 65(11), 1565-1596.

Little, R. J., Rubin, D. B. (2002). Statistical analysis with missing data.

Mashman, W. (2011). The iPad in Cardiology. JACC: CARDIOVASCULAR INTERVENTIONS, 4(2).

McInnes, I. B., Schett, G. (2011). The Pathogenesis of Rheumatoid Arthritis. N Engl J Med, 365, 2205-19.

Nakai, M., Ke, W. (2011). Review of the Methods for Handling Missing Data in Longitudinal Data Analysis. International Journal of Mathematical Analysis,

Napier, R., Kumar, M. (2013). iOS 6 Programming. Pushing the Limits. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Norman, D. A., Nielsen, J. (2010). Gestural interfaces: a step backward in usability. interactions, 17(5), 46-49.

Ording, B., Forstall, S., Christie, G., Lemay, S. O., Chaudhri, I. (2010). U.S. Patent No. 7,812,826. Washington, DC: U.S. Patent and Trademark Office.

Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B. (1996). LifeLines: visualizing personal histories. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 221-227). ACM.

Rosling, H., Rosling, R. A., Rosling, O. (2005). New software brings statistics beyond the eye. Statistics, Knowledge and Policy: Key Indicators to Inform Decision Making. Paris, France: OECD Publishing, 522-530.

Schoenberg, I., Gotlib, P., Schoenberg, R., Sherlin, H. (2001). U.S. Patent No. 6,322,502. Washington, DC: U.S. Patent and Trademark Office.

Simonic, K. M., Holzinger, A., Bloice, M., Hermann, J. (2011, May). Optimizing long-term treatment of rheumatoid arthritis with systematic documentation. In Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on (pp. 550-554). IEEE.

Smolen J.S., Landewe R., Breedveld F. C., Dougados M., Emery P., Gaujoux-Viala C., Gorter S., Knevel R., Nam J., Schoels M., Aletaha D., Buch M., Gossec L., Huizinga T., Bijlsma J. W., Burmester G., Combe B., Cutolo M., Gabay C., Gomez-Reino J., Kouloumas M., Kvien T. K., Martin-Mola E., McInnes I., Pavelka K., van Riel P., Scholte M., Scott D. L., Sokka T., Valesini G., van Vollenhoven R., Winthrop K. L., Wong J., Zink A., van der Heijde D., "Eular recommendations for the management of rheumatoid arthritis with synthetic and biological disease-modifying antirheumatic drugs", Ann Rheum Dis, vol. 69, no. 6, pp. 964– 75, Jun 2010.

Schumann, H., Müller, W. (1999). Visualisierung: Grundlagen und allgemeine Methoden. Springer DE.

Swan, M. (2009). Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. International journal of environmental research and public health, 6(2), 492-525.

Stevens, S. S. (1946). On the theory of scales of measurement.

Tufte, E. R., Graves-Morris, P. R. (1983). The visual display of quantitative information (Vol. 2). Cheshire, CT: Graphics press.

UIGestureRecognizer Class Reference (2013). https://developer.apple.com/library/ios/documentation/uikit/reference/UIGestureRecognizer_Class/UIGestureRecognizer_Class.pdf

Wallace, J. R., Scott, S. D., MacGregor, C. G. (2013). Collaborative sensemaking on a digital tabletop and personal tablets: prioritization, comparisons, and tableaux. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 3345-3354). ACM.

Wang, T. D., Wongsuphasawat, K., Plaisant, C., Shneiderman, B. (2010). Visual Information Seeking in Multiple Electronic Health Records: Design Recommendations and a Process Model.

Walny, J., Lee, B., Johns, P., Riche, N. H., Carpendale, S. (2012). Understanding Pen and Touch Interaction for Data Exploration on Interactive Whiteboards. IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 18(12).

Warner, R. M. (1998). Spectral analysis of time-series data. Guilford Press.

Weber, M., Alexa, M., Müller, W. (2001). Visualizing Time-Series on Spirals. In Infovis (Vol. 1, pp. 7-14).

Wigdor, D., Wixon, D. (2011). Brave NUI world: designing natural user interfaces for touch and gesture. Elsevier.