Fachhochschule
Salzburg University
of Applied Sciences

Implementation of a Collaborative Web Application for Annotating Gameplay Videos Based on Biometric Player Data

BACHELOR THESIS

Student Mike Schäkermann, 1110601029
Supervisor Prof. Dr. Lennart Nacke

Salzburg, May 4, 2014

Eidesstattliche Erklärung

Hiermit versichere ich, **Mike Schäkermann**, geboren am **29.08.1990** in **Dortmund**, dass ich die Grundsätze wissenschaftlichen Arbeitens nach bestem Wissen und Gewissen eingehalten habe und die vorliegende Bachelorarbeit von mir selbstständig verfasst wurde. Zur Erstellung wurden von mir keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Ich versichere, dass ich die Bachelorarbeit weder im In- noch Ausland bisher in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der den BegutachterInnen vorgelegten Arbeit übereinstimmt.

Salzburg, am **04.05.2014**

Unterschrift

Mike Schäkermann	1110601029
Vorname Familienname	Personenkennzeichen

Kurzfassung

Vor- und Zuname:	Mike SCHÄKERMANN
Institution:	FH Salzburg
Studiengang:	Bachelor MultiMediaTechnology
Titel der Bachelorarbeit:	Implementation of a Collaborative Web Application for Annotating Gameplay Videos Based on Biometric Player Data
Begutachter:	Prof. Dr. Lennart Nacke

In der vorliegenden Arbeit wird untersucht, wie Biosignale als Metrik zur Evaluierung von Videospielen verwendet werden können. In den vergangenen Jahren hat sich eine zunehmende Anzahl von Studien aus dem Bereich Games User Research (GUR) mit der Frage befasst, inwiefern Biosignale als Echtzeit-Messwerte zur Quantifizierung emotionaler Komponenten der Spielerfahrung (z.B. Erregtheit und emotionale Valenz) eingesetzt werden können. Jedoch werden nur in wenigen Studien Möglichkeiten thematisiert, wie Biosignale visualisiert werden können, um Spieleforschern aufschlussreiche Informationen intuitiv zugänglich zu machen. Vor diesem Hintergrund wird in dieser Arbeit untersucht, (1) wie psychophysiologische Signale und Gameplay-Videos visuell kombiniert werden können, um die Evaluierung von Video-Spielen zu vereinfachen und (2) wie existierende Web-Technologien genutzt werden können, um eine Software-Anwendung für Spieleforscher zu implementieren, welche online, kollaborativ und skalierbar eingesetzt werden kann. Zur Beantwortung dieser Forschungsfragen wird relevante Forschung aus den Bereichen GUR und der Visualisierung von Spielerdaten vorgestellt. Des Weiteren wurde im praxiszentrierten Teil dieser Arbeit eine Web-Applikation entworfen und implementiert, mithilfe derer Spieleforscher auf der Grundlage von Biosignalen Gameplay-Videos gemeinschaftlich annotieren können. In dem vorliegenden Prototypen werden Biosignale als Zeitreihen-Graphen visualisiert, welche durch Anlegen von Schwellenwerten interaktiv gefiltert werden können. Anhand des Beispiels des elektrischen Hautleitwerts wird aufgezeigt, wie Zeitabschnitte erhöhter emotionaler Erregung und zeitlich damit zusammenfallende Ereignisse im Spiel identifiziert und annotiert werden können. Funktionalität und Implementierungsdetails der Software-Anwendung werden eingehend beschrieben. Abschließend werden Anwendungsszenarien für die Auswertung von emotionszentrierten und rasant verlaufenden Spielen sowie von subtilen audiovisuellen Effekten entwickelt. Darüber hinaus wird auf die Erleichterung der Zusammenarbeit zwischen räumlich getrennten Spieleforschern hingewiesen, um den Beitrag dieser Arbeit für die GUR-Gemeinschaft hervorzuheben.

Schlagwörter: *Games User Research, Psychophysiologie, Biosignale, Visualisierung, Gameplay Experience, kollaborative Web-Applikation*

Abstract

In this thesis, the focus of research is on the use of biosignals as metrics for the evaluation of video games. Over the last few years, an increasing number of studies in the field of games user research (GUR) have addressed the use of biometrics as a real-time measure to quantify aspects gameplay experience (e.g., emotional valence and arousal). However, only few studies explore possibilities to visualize biometrics in a way that yields meaningful and intuitively accessible insights for games user researchers. Against this background, it will be examined, (1) how psychophysiological signals and gameplay videos can be visually combined to facilitate the process of video game evaluation and (2) how existing web technologies can be extended to create a software application for games user researchers, that works online in a large, collaborative setting. In response to these research questions, prior work from the fields of GUR and player data visualization will be reviewed. Additionally, a web application was designed and implemented, allowing games user researchers to collaboratively annotate gameplay videos, based on the visualization of biosignal time series. In the prototype, biosignals are represented as time series graphs that may be interactively drilled down and filtered to extract interesting patterns. Using the example of electrodermal activity (i.e., the electric conductance level of the skin), it will be illustrated how time intervals, featuring high levels of emotional arousal, as well as the corresponding gameplay video segments, may be identified and annotated. The application's functionality will be discussed in depth and implementation details will be given. Finally, the contribution of this work will be highlighted on the basis of multiple usage scenarios, including the evaluation of emotion-centered and fast-paced games and subtle effects of audiovisual ambience as well as the cooperation among geographically remote GUR teams.

Keywords: *games user research, psychophysiology, biometrics, visualization, gameplay experience, collaborative web application*

Contents

Acronyms

Ω ohm. 9, 50

2D two-dimensional. 12, 13, 50

3D three-dimensional. 13, 50

ANS autonomous nervous system. 9, 50

BioSt Biometric Storyboard. 5, 12, 15–17, 20, 22, 36, 40, 50

CS corrugator supercilii. 10, 50

CSS Cascading Style Sheets. 24, 28, 50

DDP dynamic data protocol. 29, 50

DOM document object model. 28, 50

e.g. exempli gratia = for instance. 50

EDA electrodermal activity. 5, 9–11, 16, 20–22, 37–43, 50

EEG electroencephalography. 5, 7, 11, 39, 41, 50

EMG electromyography. 5, 9–11, 16, 20, 37–41, 50

fMRI functional magnetic resonance imaging. 11, 50

FPS first-person shooter. 3, 37, 50

GIS geographic information system. 13, 50

GSR galvanic skin response. 50

GUR games user research. 3–7, 9–12, 14–16, 18–20, 36–38, 40, 41, 50

HCI human-computer interaction. 3, 5, 6, 50

HDFS Hadoop Distributed File System. 34, 50

HFAA hemispheric frontal alpha asymmetry. 50

HR heart rate. 11, 50

HTML HyperText Markup Language. 24, 27, 28, 50

HTTP HyperText Transfer Protocol. 27, 29, 30, 33, 36, 50

Hz Hertz. 30, 37, 50

i.e. id est = that is. 50

JSON Javascript object notation. 33, 50

LOD level-of-detail. 31, 33, 50

MB megabyte. 30, 50

NoSQL Not-only SQL. 25, 36, 50

OT operational transformation. 27, 50

RTMP real-time messaging protocol. 30, 50

RTP real-time transport protocol. 30, 50

S siemens. 9, 50

SCL skin conductance level. 50

SNS sympathetic nervous system. 9, 50

SQL Structured Query Language. 50

TCP Transmission Control Protocol. 27, 50

UCD user-centered design. 5, 50

ZM zygomaticus major. 10, 39, 50

1 Introduction

In the video games industry, a rapidly growing and highly competitive market, a polished gameplay experience is one of the key factors for the economic success of a game title. Therefore, the evaluation of playability and gameplay experience, as a way to inform game designers and developers about the strengths and weaknesses of a game prototype, has become an integral part in the development cycle of many video game productions (cf. Pagulayan et al. 2003 and Zammitto 2011).

The research effort of this thesis is located at the intersection of games user research (GUR) as a sub-field of human-computer interaction (HCI) and physiological computing. Specifically, the use of electrophysiological biosignals to facilitate the evaluation of gameplay videos in large, collaborative GUR settings will be the subject of investigation.

The applicability of biosignals as psychophysiological metrics (i.e., indicators for subjective states, such as emotional valence and arousal) in the process of evaluating video games has been demonstrated by both industrial (e.g., Ambinder 2011 and Chalfoun 2013) and academic research (e.g., Drachen et al. 2010 and Nacke, Grimshaw, and Lindley 2010). Psychophysiological measurement techniques have been shown to complement conventional qualitative evaluation techniques, such as post-hoc questionnaires and player interviews, by providing an unobtrusive and sensitive way to quantify emotion-relevant player data in real time (Mirza-Babaei et al. 2011).

Despite a growing body of literature about the statistical interpretation of psychophysiological player data (cf. Drachen et al. 2010, Nacke, Grimshaw, and Lindley 2010 and Wehbe et al. 2013) and the visualization of behavioral player data (cf. Wallner and Kriglstein 2014 and Drachen 2014), the visualization of psychophysiological player data still is a scarce topic of research. Instead, most studies in the field of player data visualization focus on the representation of behavioral data, acquired through in-game event logging (i.e., telemetry). In this field, a number of approaches base on the assumption that gameplay consists in navigating a character through a two- or three-dimensional environment, as it holds true for first-person shooters (FPSs) and other genres. These approaches tend to visualize behavioral player data within its spatial context, e.g., in the form of heatmaps or trajectories of player movement (cf. Drachen 2013). More abstract visualization frameworks address a broader range of game genres (e.g., also abstract puzzle games) by making the visualization independent of spatial dimensions. Instead, gameplay is represented in a more generic way, e.g., as a finite state machine, visualized as a node-link diagram (Wallner and Kriglstein 2014).

Only few visualization approaches harness the potential of psychophysiological player data to gain insights about emotional components of gameplay experience. One of the few studies, addressing this research gap, puts forward the Biometric Storyboard approach (Mirza-Babaei et al. 2013b), in which biometric and behavioral player data are combined with gameplay videos and player annotations to form a graph-like visualization of gameplay experience.

However, for the analysis of highly diverse player data, experts from different scientific backgrounds are required to combine their interdisciplinary expertise and look at patterns, comprised in the data, from various angles. For the combination of behavioral and psychophysiological player data, games user researchers, experimental psychologists and game designers need to collaborate closely in order to construct meaningful interpretations and derive actionable conclusions about improvements to the game design.

Despite the need for interdisciplinary cooperation, however, no software application has been proposed in the literature so far, enabling interdisciplinary GUR groups to visualize, inspect and annotate player data in a collaborative manner. Therefore, the objective of this thesis is to enhance the toolset and workflows of games user researchers by proposing a web-based GUR application to support the collaborative annotation of gameplay videos, based on visualizations of psychophysiological player data. In accordance with this objective, the following research questions will be addressed in this thesis both from a theoretical and a practical perspective:

1. How can psychophysiological signals and gameplay videos be visually combined to facilitate the process of video game evaluation?
2. How can existing web technologies be extended to create a software application for games user researchers, that works online in a large, collaborative setting?

In the practical part of this work, the prototype for an online, collaborative GUR tool was implemented on the basis of web technologies, harnessing the potentials of interactive vector graphics, online collaboration and portability. In the theoretical part, relevant work from the fields of GUR and player data visualization will be reviewed. Special emphasis will be placed on the role of biosignals in GUR as well as on previous approaches to visualize behavioral and psychophysiological player data. Moreover, the results of the practical part will be presented in the form of a description of the tool, a selection of implementation details as well as considerations about an optional extended server architecture. As no empirical methods are applied, the methodology of this thesis is a qualitative one, based on literature review and the design, implementation and discussion of the abovementioned software prototype.

The remainder of the thesis is structured as follows. Related work from the fields of GUR (section 2.1) and player data visualization (section 2.2) will be presented in the next section. In section 3, the prototype for an online, collaborative GUR tool will be described (section 3.1), implementation details will be given (section 3.2) and an optional server architecture for extended scalability will be proposed (section 3.3). Before the thesis concludes with a summary of the presented work and future directions (section 5), the GUR software prototype will be discussed on the basis of different usage scenarios and its specific contribution to the GUR community as well as limitations of the current implementation will be highlighted in section 4.

2 Related Work

In this section, previous work from the fields of GUR (section 2.1) and the visualization of player data (section 2.2) will be presented. It will be highlighted in what way GUR distinguishes itself from user research for productivity applications (section 2.1.1). In more detail, it will be elaborated on the role of biometrics in the context of GUR (section 2.1.2), using the examples of electrodermal activity (EDA), facial electromyography (EMG) and electroencephalography (EEG). In terms of player data visualization, a brief overview on visualization concepts for behavioral player data will be given (section 2.2.1), followed by a more detailed presentation of the Biometric Storyboard (BioSt) approach (section 2.2.2) as an example of a GUR framework, also visualizing psychopysiological player data.

2.1 Games User Research (GUR)

Games user research is a sub-field of HCI (Hewett et al. 1992), in which the interaction of users with digital games lies in the focus of attention. As a scientific discipline, it is located at the intersection of HCI, game development and experimental psychology (Seif El-Nasr et al. 2012). Recent GUR methodologies have been adapted from the classic user-centered design (UCD) (ISO-13407 1999) approach, originally conceived as an iterative design process for productivity applications, in which a strong emphasis is on the needs, tasks and environment of the application user. This section will tackle the question, in what way GUR differs from user research for productivity applications (section 2.1.1), and provide an overview on the role of different electrophysiological biosignals in the context of GUR (section 2.1.2).

2.1.1 Differentiation from User Research for Productivity Applications

Digital games are a special genre of software applications. In contrast to productivity applications, which are designed as tools to help users perform specific tasks more efficiently, games generally serve a different purpose. They are mostly considered a form of interactive entertainment or art rather than being merely functional computer programs (Pagulayan et al. 2003). As conventional user research methods are typically designed for productivity applications, Mirza-Babaei et al. 2013a derive the need for an adaptation of GUR methodologies to the specific characteristics of digital games. They allege the example of negative emotions such as frustration, typically avoided in conventional frameworks for user research, which may be an intended part of the feedback loop of a digital game (Mirza-Babaei et al. 2013a, p. 2).

Productivity applications are result-oriented applications, conceived to aid users in fulfilling self-defined tasks by removing or alleviating constraints in their work process. They are typically designed with consistency and functionality in mind and tend to avoid negative emotions. In contrast to that, video games put the user into a process of play, in which he/she typically has to achieve specific goals by overcoming built-in constraints and following specified rules. Games do not necessarily strive for consistency, but may aim at inspiring the player by generating a variety of stimuli within the game world. As sources of inspiration, games are not restricted to the generation of positive emotions, but their game loops might allow for negative emotions to be temporarily aroused in the player (cf. Pagulayan et al. 2003 and Mirza-Babaei et al. 2013a).

2.1.2 Biosignals as GUR Metrics

The use of biosignals in GUR is one of several approaches that were developed by the GUR community to gain insights about a player's subjective gameplay experience. In the context of this thesis, a biosignal is defined as an electrophysiological signal, recorded on the skin surface of a participant, although, in other contexts, the same term may refer to all quantifiable signals, originating from biological organisms. Biosignals are used as measures in psychophysiology, a scientific discipline at the intersection of psychology and physiology, in which mental (i.e., cognitive or emotional) processes are inferred from the physiological activity of a participant (Cacioppo, Tassinary, and Berntson 2007). Nacke 2013 presents a categorization scheme for GUR methods, in which methodologies are classified along two dimensions: qualitative vs. quantitative and subjective vs. objective.

For scientific methodologies, objective methods of data acquisition are generally preferred over more subjective ones due to their higher comparability and reproducibility. As a sub-field of HCI, however, GUR is strongly related to psychology and the major focus of research, gameplay experience, is an inherently subjective phenomenon. Therefore, it is logical that, for GUR methodologies, also subjective measures are of certain relevance.

Compared to qualitative measurements, quantitative data benefit from a completely or at least partially automatable data preparation and evaluation process and the possibility to compare different observations numerically. On the other hand, qualitative data offers a higher degree of flexibility, because there is no defined scheme to be matched. Instead, also unanticipated observations may be registered, when data is acquired qualitatively.

Figure 1 shows a collection of GUR methods, arranged in line with the abovementioned scheme (Nacke 2013). According to this classification, physiological measures belong to the most objective and quantitative evaluation techniques. Correspondingly, in a recent review, psychophysiological methods are characterized as an "objective, continuous, real-time, non-invasive, precise, and sensitive way" to register emotional components of gameplay experience (Kivikangas et al. 2010).

Figure 1: An overview of game user research methods grouped together by similarity in a quantitative or qualitative and objective or subjective focus based on Mandryk 2008 (reprinted from Nacke 2013, 586)

It is important to mention, however, that psychophysiological methods are usually viewed as an extension to, rather than a substitute for conventional observation-based user research methods (cf. Mirza-Babaei et al. 2011 and Hakvoort et al. 2011). In a recent attempt to assess the contribution of psychophysiological game evaluation techniques, biometrics are stated to complement observation-based techniques, especially by exposing "latent issues [...] related to players' feelings, immersion and gameplay experience" (Mirza-Babaei et al. 2011). For certain event types, biometric methods are even reported to reveal up to 63% more game design issues than observation-only techniques (Mirza-Babaei et al. 2011). On the downside, the need for expensive, specialized recording devices, precisely controlled experiments and relatively large participant samples is identified as the major drawback of psychophysiological methods (cf. Kivikangas et al. 2010).

In this section, a two-dimensional model for the classification of emotions, called the circumplex model of affect (Russell 1980), will be introduced, which is applied by a number of GUR studies to quantify emotional components of gameplay experience. Moreover, three electrophysiological measures that have been successfully used as evaluation metrics for gameplay experience are presented in more detail: electrodermal activity, facial electromyography and EEG. The three measurement techniques will be described, their benefits and drawbacks will be identified and it will be given an overview on GUR studies, in which the techniques have been applied.

Circumplex Model of Affect

The circumplex model of affect, introduced by Russell 1980, is a model for the classification of emotions along the following two dimensions:

- Arousal
- Valence

Arousal refers to the intensity of an emotion. Therefore, emotional states of high activation, such as excitement, exhibit high arousal scores, whereas emotions with a low activation level, such as fatigue, are rated low on the arousal scale. Valence, on the other hand, is a term to describe whether an emotion is positive or negative. Therefore, pleasant emotions like happiness are rated high in valence, whereas unpleasant emotions like sadness have low valence scores. Figure 2 illustrates the two dimensions of the circumplex model of affect, mapping several emotions to the valence (horizontal) and the arousal (vertical) scales.

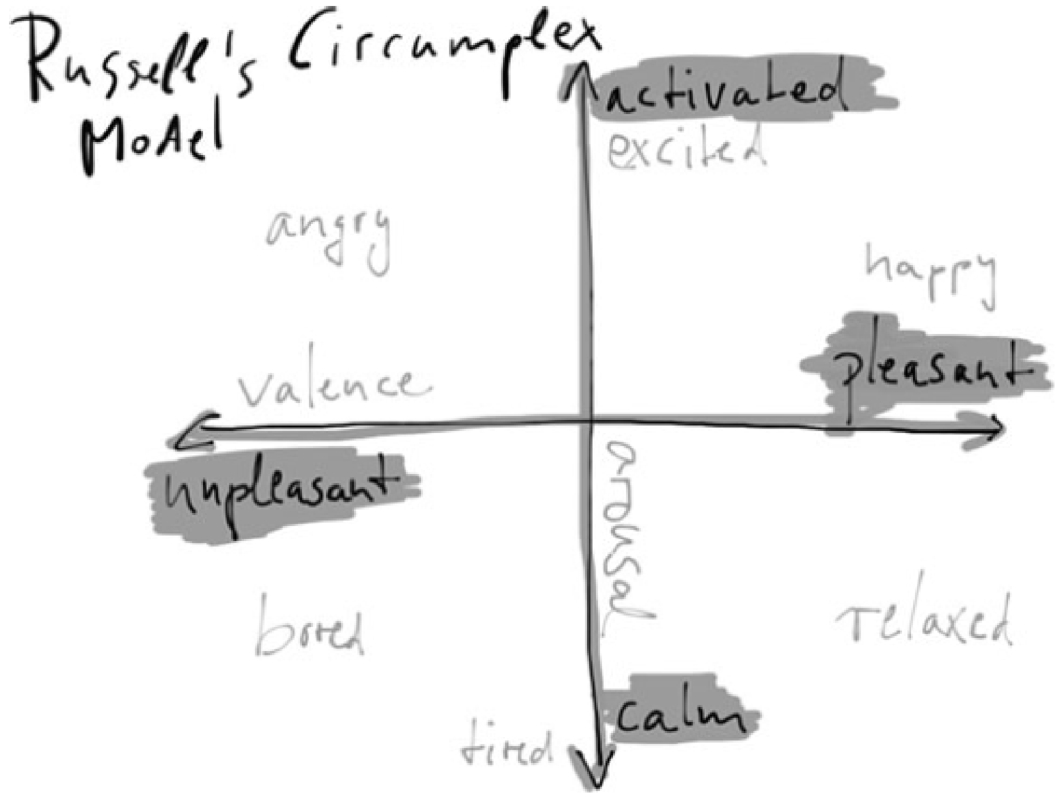

Figure 2: Two emotion dimensions (valence and arousal) in the circumplex model from Russell 1980 (reprinted from Nacke 2013, 600)

Electrodermal Activity

Electrodermal activity is a term for the electric conductance of the skin. The inverse of electric conductance is electric resistance. Therefore, EDA values are either expressed in the unit siemens (S) for conductance or in its reciprocal unit ohm (Ω) for resistance (Cacioppo, Tassinary, and Berntson 2007, p. 162). The direct physiological basis for a change in EDA is the secretory activity of eccrine sweat glands, located on the surface of the skin (Birbaumer and Schmidt 2005, p. 224). EDA is usually recorded at skin regions with a particularly high density of sweat glands. The hands' palmar and feet's plantar parts of the skin belong to these regions of high gland density. This is one of the reasons why they are often used for EDA measurements (Cacioppo, Tassinary, and Berntson 2007, p. 160).

The main function of most eccrine sweat glands is thermoregulation. It could be shown, however, that the activity of the sweat glands on the palmar and plantar surfaces is more closely related to psychologically significant stimuli than to thermal stimuli (Cacioppo, Tassinary, and Berntson 2007, p. 160). This is another reason why hands and feet are the regions, preferably used for EDA measurements in the context of psychophysiological experiments. Eccrine sweat glands are activated by sympathetic nerval endings through the neurotransmitter acetylcholine. These nerve cells belong to the sympathetic nervous system (SNS). The SNS is the division of an organism's autonomous nervous system (ANS) that mediates an increase of the organism's overall state of alert and activates its fight-or-flight response when exposed to physical or mental stress (Klinke and Bauer 2005, p. 789).

Several GUR studies apply methodologies, in which EDA is used as an index for the participant's level of arousal (cf. Nacke and Lindley 2008, Nacke, Lindley, and Stellmach 2008 and Nacke, Grimshaw, and Lindley 2010). EDA is often combined with other measures like facial EMG that yield information about the participant's emotional valence to derive a two-dimensional profile of emotion in accordance with the circumplex model of affect (Russell 1980).

The recording of EDA levels is a cheap and uncomplicated measuring technique, as it only involves - at least in the most simplistic setting - two electrodes to be attached to the distal or intermediate segment of one of the participant's fingers (see figure 3, right side). Compared to other biosensors, such as facial EMG, the measurement of EDA is less intrusive. Moreover, the interpretation of EDA may be considered straightforward, as only two data dimensions (conductance level and time) need to be taken into account (Nacke 2013). It is often required, however, due to varying average EDA levels among participants, that the recorded EDA data is normalized before data interpretation takes place (Nacke 2013). In addition, EDA levels are reported to start drifting after a period of time, which is mainly attributed to the overall increase of sweat production during a gameplay session. Therefore, Nacke 2013 recommends to incorporate resting periods between different gameplay sessions to compensate for this potential confounder.

Facial Electromyography

Facial EMG is a psychophysiological technique to measure the electric activity of facial muscles over time. It involves the attachment of non-invasive electrodes to the skin surface of a participant's face. Figure 3 (left side) shows a participant with two facial EMG electrodes, attached above his eyebrows and to his cheek respectively.

Figure 3: EMG sensors (left) and SC sensors (right) (reprinted from Mirza-Babaei et al. 2013b, p. 3)

The generation of basic human emotions is correlated with the activation of certain facial muscles (Birbaumer and Schmidt 2005). In GUR, it is therefore a common approach to measure the activity of muscles, involved in the generation of smiles and frowns (Mirza-Babaei et al. 2011). More specifically, the activity of corrugator supercilii (CS), the frowning muscle, is used as a metric for negative valence, whereas the activation level of zygomaticus major (ZM), one of the smiling muscles, is taken as an indicator of positive valence (cf. Nacke and Lindley 2008, Nacke, Grimshaw, and Lindley 2010, Hakvoort et al. 2011 and Mirza-Babaei et al. 2011).

As for all electrophysiological measurement techniques, facial EMG features a relatively high temporal resolution as well as a high sampling precision, allowing it to register very quick and fine-grained changes in facial muscular activation. Moreover, other than EDA, emotional variations are instantaneously reflected in a change of facial muscle activity (Cacioppo, Tassinary, and Berntson 2007), making it easier for researchers to associate the eliciting in-game events to fluctuations in the facial EMG signal. On the other hand, facial EMG is sensitive to noise, which may either be caused by an inadequate measurement setup (e.g., poor contact between skin and electrode) or originate from untargeted muscular activity (cf. Kivikangas et al. 2010). Apart from that, facial EMG is considered a rather intrusive technique, compared to EDA measurements, as it requires electrodes to be attached to the participant's forehead and cheek regions. Finally, EMG recording devices are relatively expensive, compared to other techniques, such as EDA (Nacke 2013).

Electroencephalography

EEG is an electrophysiological brain imaging technique. It is based on the measurement of electric potentials at different scalp regions over time. Each EEG electrode outputs a voltage graph, based on the electric potential of the scalp region it is attached to (Cacioppo, Tassinary, and Berntson 2007). EEG recordings allow researchers to draw inferences about the electric activity of different areas of the cerebral cortex, generated by the cumulated activity of its underlying neural populations. The cerebral cortex is the outermost part of a human's brain, located closest to the cranial bone and the scalp. An extensive body of literature from the fields of psychology and neurology attribute perceptive, cognitive and emotional functions to this region of the brain (cf. Birbaumer and Schmidt 2005, Cacioppo, Tassinary, and Berntson 2007, Klinke and Bauer 2005 and Luck 2005), making EEG a potentially interesting measure for games user researchers to infer cognitive and emotional states, such as attention, boredom and arousal, in real time.

However, to this date, the use of EEG in GUR has been scarce. Kivikangas et al. 2010 hypothesize that this may be due to the "complicated nature of the signal", emphasizing the methodological complexity, introduced when EEG is combined with a complex stimulus such as a digital game. Two examples of the use of EEG in GUR are the studies of Wehbe et al. 2013 and Nacke and Lindley 2008. Wehbe et al. 2013 propose an EEG-based approach to assess the efficiency of in-game learning. In the study, the effect of imitation learning on performance in video games is investigated. This is done by comparing gameplay sessions, that were preceded by the presentation of a gameplay video, to those, performed without the tester first watching a gameplay video before playing. Nacke and Lindley 2008 apply EEG in combination with heart rate (HR), facial EMG and EDA to make assumptions about a player's flow and immersion scores and test these assumptions against the player's subjective scores, gained from questionnaires. The authors state a positive correlation between the subjective and objective metrics, concluding that "real-time emotional profiles of gameplay" (Nacke and Lindley 2008, p. 1) may be constructed by combining these metrics.

Compared to other brain imaging techniques, such as functional magnetic resonance imaging (fMRI), one drawback of EEG is its relatively low spatial resolution. As mentioned above, the output of each EEG electrode results from the accumulated electric potential of a large number of neurons, measured indirectly through a stack of four anatomical layers, which are (from inside to outside): two meninges, the cranial bone and the scalp (Silbernagl and Despopoulos 2007). This drawback may be partially alleviated by the attachment of additional electrodes. However, the maximum number of electrodes that can be used for EEG measurements is about 256 (Cacioppo, Tassinary, and Berntson 2007). On the other hand, a benefit of EEG is that it features the highest temporal resolution among all brain imaging techniques with sampling rates between 2048 and 4096 Hz (Cacioppo, Tassinary, and Berntson 2007). Moreover, in contrast to most other biosignals, a single EEG recording contains enough information to derive multiple biometrics for different cognitive and emotional dimensions from it (Cacioppo, Tassinary, and Berntson 2007).

2.2 Visualization of Player Data

One common GUR technique, used to track players' actions inside a virtual game world, is telemetry. The term refers to the process of logging in-game events or game states to a datastore (e.g., by transferring the data to a remote database or by writing it to a local file). The resulting event logs may be used to identify, e.g., player segments (Drachen et al. 2010) or interesting patterns in the sequences of player actions (Wallner and Kriglstein 2014). Telemetry presents an unobtrusive way of collecting behavioral player data, which may be used by game designers and games user researchers to draw inferences about how players behave inside a game world. It is used to derive actionable conclusions for enhancements to the game (e.g., bug fixes or adaptations of the game mechanics) (cf. Drachen 2013).

As a number of video games exhibit highly complex game worlds and rule systems, potentially handling massive amounts of players (e.g., massive multiplayer open world genres), telemetry may yield an extensive body of highly diverse event data. In these cases, the resulting data sets cannot be interpreted in its raw form any longer, requiring techniques to visualize the contained information in a lucid way. Moreover, for the interpretation of raw event logs or purely numeric aggregative statistics, expert analysts are needed to inform game designers and developers about helpful information, comprised in the data.

Therefore, in order to make large and highly diverse sets of behavioral player data intuitively accessible for game designers, a number of visualization approaches have been developed by the GUR and information visualization communities, including heatmaps, node-link diagrams, matrix representations and others. Despite the use of psychophysiological player data as a real-time measure for the assessment of player emotions, only little research has been conducted about the visualization of biometrics to inform game designers and games user researchers about emotionally relevant elements in the design of a game. One of the few attempts to visualize psychophysiological data in the context of GUR is the BioSt framework, presented by Mirza-Babaei et al. 2013b. This section will offer a short overview on previous attempts to visualize behavioral player data (section 2.2.1) and provide an introduction to the BioSt approach (Mirza-Babaei et al. 2013b) (section 2.2.2).

2.2.1 Visualization of Behavioral Data

One of the simplest visualization techniques, used to analyze behavioral player data, is the heatmap. Heatmaps are static images, in which the color of each pixel codes the value of a specified variable at a certain position in space. In game analytics, heatmaps usually represent the two-dimensional (2D) view on a game interface (e.g., the menu or an in-game level), visualizing variables, such as the number of death events at certain positions in the game world or the number of clicks at certain areas in the game menu (cf. Drachen 2013, pp. 365 - 369).

The major restriction of heatmaps is that they visualize only one variable at a time, hampering the interpretation of multiple gameplay metrics in a shared context. To compensate for this issue, multi-layer maps, inspired by geographic information systems (GISs), have been proposed, in which multiple gameplay metrics or aggregates of those (e.g., player trajectories, death events, respawn events and others) are visualized in the same spatial context, making use of semi-transparency. Interactive implementations of these multi-layer maps allow analysts to perform filter and drill-down processes on spatial event data to investigate only specific aspects of player behavior.

Visualization approaches, relying on the representation of spatial data, however, are based on the assumption that gameplay takes place by navigating a character through a virtual environment. Therefore, these types of visualization cannot be applied to games of more abstract genres, in which event data are not necessarily associated with positions in a 2D or three-dimensional (3D) space, such as abstract puzzle games.

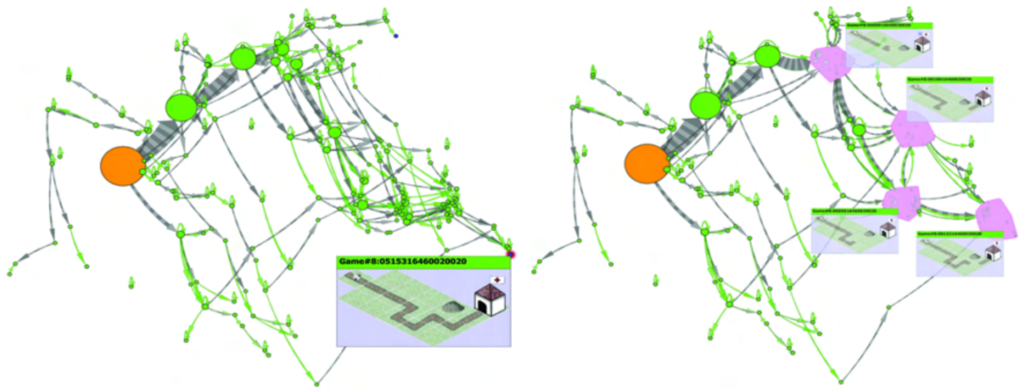

Figure 4: Left: Visualization of Level 8 from DOGeometry. Game states with less than three visits were hidden to reduce the visual complexity of the graph. The visualization calls attention to the upper right area from the start state (orange). The first two moves seem to be clear but then the cluttering indicates confusion in how to proceed correctly. Right: Clustering these game states helps to identify the principal way through this area. A cluster is represented by the screenshot associated with the most frequently visited game state. (reprinted from Wallner and Kriglstein 2012, p. 6)

To tackle this shortcoming, Wallner and Kriglstein 2012 have developed a method to represent gameplay as a finite state machine, in which players navigate between different game states, using actions. In this approach, in-game event data is visualized as a node-link diagram, in which nodes refer to game states and links to player actions. The abstract formalism of a finite state machine is applied in order to keep the visualization technique independent of spatial dimensions. This additional layer of abstraction ensures that the visualization approach can be applied to any game, independent of its specific genre. Figure 4 juxtaposes two node-link diagrams, based on the same in-game event data from a puzzle game called *DOGeometry*. The left diagram was created without prior clustering, whereas, for the right diagram, the original event data was clustered in order to remove visual clutter. This clustering technique can be used to help games user researchers identify the principal sequences of actions, chosen by most players to navigate through the game level.

All of the visualization approaches, presented so far, have in common that they visualize only one set of player data, lacking the possibility to compare two different data sets. However, in GUR settings, where the overall objective is to enhance the gameplay experience for users, the evaluation of changes in player behavior is of high interest. Therefore, it is a desirable feature for games user researchers to automatically compare how players behave in different player segments or in different versions of a game. In response to this, Wallner and Kriglstein 2012 introduced the concept of difference graphs in an improved version of their visualization framework.

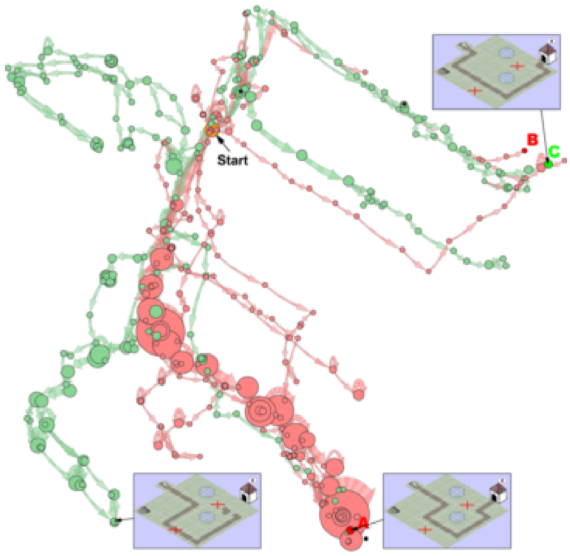

Figure 5: Graph showing the differences between 8- and 9-year-old players. Reddish parts were more frequented by 8-year-olds and greenish parts by 9-year-olds. States corresponding to solutions are shown in more saturated colors. Nodes with less than 10% change are hidden. For some nodes the associated configuration of the grid is shown (red X mark the location of bones) (reprinted from Wallner 2013, p. 5)

Difference graphs allow game designers to analyze changes in player behavior, introduced in a game over time (e.g., through updates or patches), or to compare the in-game behavior of players with different demographics (e.g., gender or age). Only the differences between the two data sets, being compared, are highlighted and those nodes, for which the difference between the two data sets is negligible, are hidden from the diagram. Figure 5 depicts an example of a difference graph, visualizing the different preferences in solving a specific level in *DOGeometry* between groups of 8- and 9-year-old children. In particular, the figure highlights that 8-year-olds tend to favor a different way of solving the level than 9-year-olds, helping game designers and analysts to define more profound and meaningful player segments. In a more recent revision of the same approach (Wallner and Kriglstein 2014), the authors present further enhancements like subgraph matching, allowing analysts to query the play graph for specific patterns, and a path explorer to filter and sort graphs for specific criteria. Furthermore, a variety of additional diagram types were introduced, such as matrix representations for complex graph structures, path visualizations and heatmaps.

2.2.2 Biometric Storyboards

Mirza-Babaei et al. 2013b propose a GUR framework, called Biometric Storyboards, in which psychophysiological measures are used in combination with other types of user data. The BioSt approach is based on the concept of graph storyboarding, in which graph-like visualizations are used to communicate user experience. The integration of gameplay video recordings, subjective player comments and objective biosignal data as well as observational notes and in-game metrics results in a multi-facette representation of gameplay experience. Moreover, the approach provides a way to compare this visualization of the recorded data to the gameplay experience as originally intended by the game designer. The BioSt framework defines the following three data views, which will be covered in this section:

1. the intended player experience graph
2. the player's input view
3. the GUR view

Intended Player Experience Graph

The intended player experience graph (see figure 6) is used by the game designer to sketch the anticipated gameplay experience as a curve to indicate different levels of excitement during gameplay. The tool allows it to subdivide the gameplay session into smaller time segments, referred to as *game beats*, which share game events of common thematic areas (Mirza-Babaei et al. 2013b, p. 3).

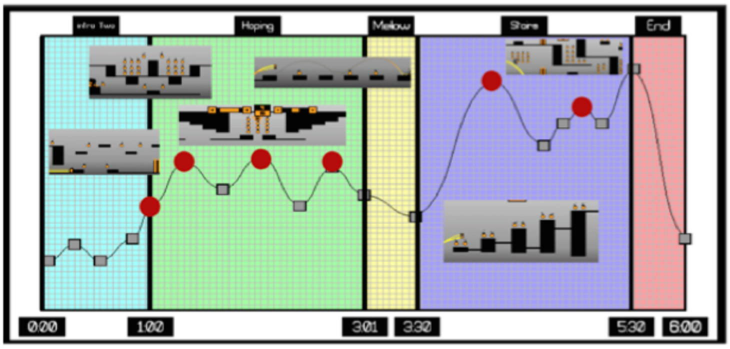

Figure 6: Intended player experience graph (representing what designers think exciting gameplay moments are) showing game beats*, times and key events (reprinted from Mirza-Babaei et al. 2013b, p. 3)*

Player's Input View

The player's input view is used by the player and researcher together to search the gameplay video for sections, relevant for improvements to the game, and insert player comments at the corresponding time positions. The player may have comments marked as positive or negative, according to the valence his/her experience in the corresponding gameplay situation (Mirza-Babaei et al. 2013b, p. 3).

GUR View

The GUR view establishes the link between the intended player experience graph and the actual player data, represented by the single player's data graph (see figure 7) and the aggregated experience graph (see figure 8).

The single player's data graph, shown in figure 7, displays four dimensions of player data along a timeline (from top to bottom):

1. normalized EDA level graph
2. bars for above-threshold segments of the facial EMG signals (green for smiling muscle and red for frown muscle)
3. player comments, indicated by stickmen symbols (blue for positive and red for negative)
4. time, spent in different *game beats*, indicated by color-coded lines

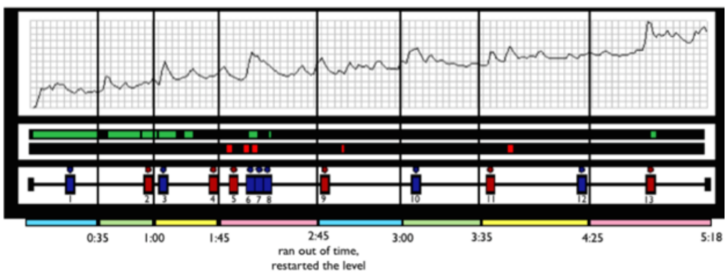

Figure 7: Single player's data graph in GUR view synchronized based on the frame counter timestamp (reprinted from Mirza-Babaei et al. 2013b, p. 3)

The aggregated data from all players is visualized in the form of the GUR aggregated player experience graph, shown in figure 8. Matching the visual format of the intended player experience graph, it is meant to facilitate the comparison between the intended and actual levels of excitement (arousal) and the average time periods, spent in each *game beat*.

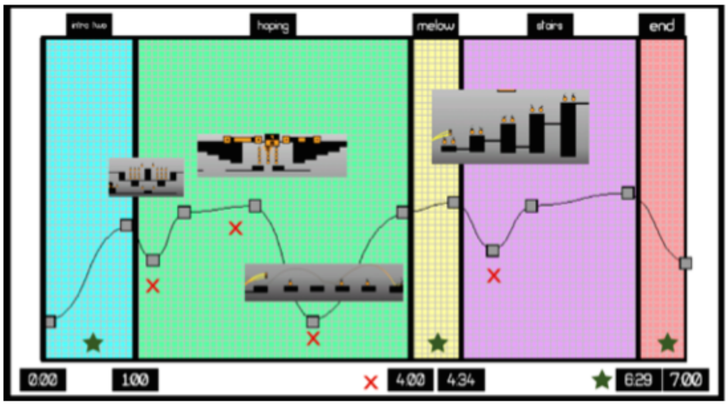

Figure 8: GUR aggregated player experience graph, indicating areas of difficulty and average time spent in each game beat *(reprinted from Mirza-Babaei et al. 2013b, p. 4)*

In the study, the effectivity of the BioSt approach is evaluated in comparison to classic user testing methods on the one hand and the absence of user testing on the other hand. The results of the study indicate that BioSt and classic user testing perform considerably better than no user testing at all, resulting in game prototypes that are rated significantly better by players (Mirza-Babaei et al. 2013b). BioSts are stated to help provide "more nuanced design suggestions" (Mirza-Babaei et al. 2013b, p. 2) than classic user testing, resulting in a higher quality of gameplay experience and motivating more fine-grained improvements to game mechanics.

The BioSt approach was taken as inspiration for the practical part of this thesis, in which a web-based, collaborative software application for games user researchers was designed and implemented, allowing researchers to annotate gameplay videos, based on visualizations of psychophysiological signals. The functionality and implementation details of the tool will be described in the following section.

3 Prototype for an Online Collaborative GUR Tool

In the practical part of this thesis, an application was implemented, allowing games user researchers to annotate gameplay videos on the basis of biometrics, by combining psychophysiological data and gameplay video recordings into an interactive visualization of gameplay experience. The result is a prototype for an online collaborative GUR tool, which can be used, e.g., in the local network of an industrial or academic GUR laboratory to accelerate the process of evaluating gameplay videos, based on psychophysiological data. The name of the application, *Repidly*, reflects the underlying idea of providing an aid for games user researchers to create GUR reports more rapidly. Zammitto et al. 2014 express the relevance of this endeavor, emphasizing the importance of effective reporting procedures for GUR results in both academic and industrial contexts.

In the following sections, the application's functionality and its visualization concepts will be described (section 3.1), followed by a presentation of relevant implementation details, regarding platform and software design decisions as well as collaboration features and random access to player data (section 3.2). Concludingly, considerations about an optional extended server architecture will be put forward to support large-scale, parallelized data processing in a scalable setup (section 3.3).

3.1 Description of the Tool

The application, developed in the pratical part of the thesis, is a data visualization and inspection tool for games user researchers. It is designed to enable researchers to combine video recordings, psychophysiological measurements and subjective annotations into one coherent data view. Biosignal time series may be drilled-down and filtered interactively in order to extract and annotate emotionally relevant segments of the corresponding gameplay videos. The overall objective of the tool is to allow games user researchers to approach player data from a psychophysiological perspective in cooperation with other researchers. The approach is based on the assumption that the exchange between researchers from different backgrounds, such games user researchers, games analysts and experimental psychologists, yields deeper insights into and more meaningful interpretations of diverse player data. In this section, it will be described in detail, how data is uploaded and grouped in reports (section 3.1.1), how psychophysiological signals and gameplay videos are visually combined for an interactive exploration of the data (section 3.1.2) and how gameplay situations may be annotated in a collaborative manner (section 3.1.3).

3.1.1 Report Creation and Data Upload

As *Repidly* is designed to aid games user researchers in the collaborative creation of GUR reports, the topmost instances in the organizational structure of *Repidly*'s data model are reports. They are used to group data, belonging to the same game prototype or evaluation procedure, and to manage the access rights of different users. If a user creates a report, he/she will be registered as its owner, granting full access rights to all its contents and configurations. Within a report, data is organized in test sessions, which in turn consist of one ore more testers. Figure 9 presents the graphical user interface, provided to add testers to a test session. The possibility to add multiple testers to a single test session was introduced in order to support the evaluation of multiplayer games, in which multiple testers and their corresponding data are associated with a single test session.

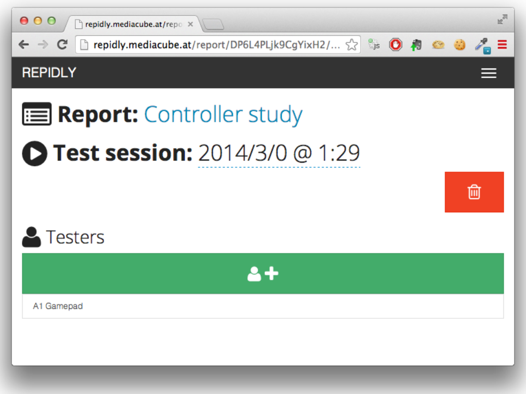

Figure 9: The graphical user interface to create a new tester for a test session

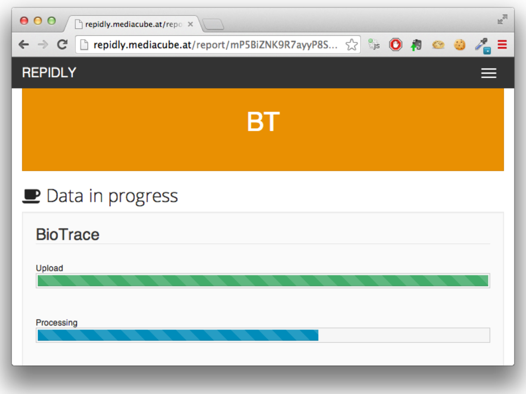

Figure 10: Two progress bars, providing feedback about the upload and processing progress of inserted file data

For every tester, researchers may upload video and biosignal data by dragging a video or biosignal file onto the corresponding upload button. Subsequently, the file data is transferred over the network, saved on the server and post-processed automatically. For video files, a conversion to web-compatible, streamable formats is performed. Biosignal files, on the other hand, are parsed and searched for time series data. The biosignal channels, detected in a file, are inserted into the database and post-processed to be available at multiple resolutions (for further details, see section 3.2.4). While a file is uploaded to and subsequently processed on the server, the user interface provides a real-time feedback in the form of two separate progress bars for the upload and processing procedures respectively (see figure 10). The two bars are intentionally kept separate, as they have different implications for the user. On the one hand, the completion of the uploading process indicates that the web application may be closed, because all required data has been transferred to the server, whereas the completion of the processing procedure implies that the uploaded data has been successfully prepared for visualization and inspection.

3.1.2 Visualization and Data Exploration

As previously mentioned, in *Repidly*, biosignal time series and video recordings may be combined in a coherent data view, inspired by the BioSt concept (Mirza-Babaei et al. 2013b). Specifically, the single player's data graph (as part of the BioSt GUR view, see section 2.2.2) serves as a template for the data visualization in *Repidly*. Both the BioSt single player's data graph and the *Repidly* visualization present biosignal time series graphs and comments along a timeline.

In the BioSt framework, gameplay videos are watched chronologically as a whole to extract gameplay intervals, relevant for game design enhancements. Based on the time positions of the extracted segments, player comments are inserted into the timeline of the single player's data graph. In contrast to that, the visualization in *Repidly* allows for an interactive inspection of the video recordings in non-chronological order, based on the position of interesting patterns, comprised in the biosignal data. To take two examples, EDA signals may be used to extract time intervals, based on a player's level of sympathetic activation (corresponding to emotional arousal), whereas facial EMG recordings of the smiling and frowning muscles may be used to identify time intervals of increased or decreased emotional valence.

All biosignals and video recordings, associated with a certain tester, are accumulated in the data visualization view, in which biosignals are visualized as time series graphs along a timeline. Each time series graph may be interactively drilled down, using a time range selector, placed in a low-resolution overview of the entire graph at the top of the page. Figure 11 shows the time series graph of an EDA channel. A dropdown menu offers the user the possibility to choose between three different biosignal channels to be visualized. In figure 12, the same principle is applied to video recordings. A dropdown menu presents previews for two different video channels, which can be activated or deactivated for inspection.

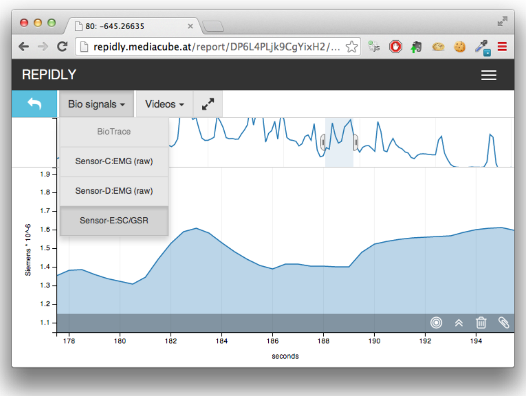

Figure 11: Selection of three different biosignal channels in Repidly*'s data visualization view*

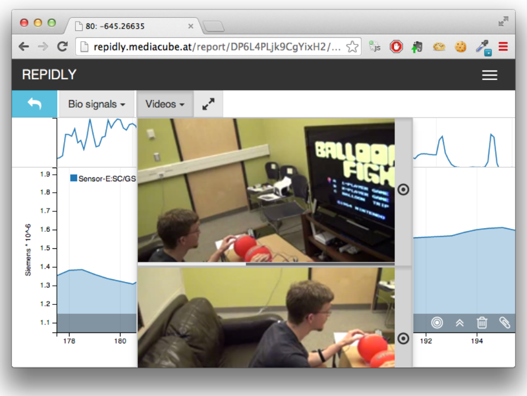

Figure 12: Selection of two different video channels in Repidly*'s data visualization view*

To identify time intervals with amplitudes above or below a certain value, a threshold bar may be positioned in the time series graph. Figure 13 depicts the threshold bar, used to extract time segments with amplitudes above the specified threshold value. As other biosignals may exhibit an converse polarity, the filter mechanism may be inverted to highlight those time intervals with below-threshold amplitudes. The result is illustrated in figure 14.

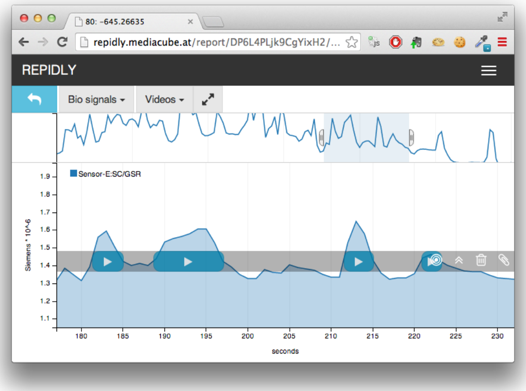

Figure 13: Identification of time intervals, in which the EDA signal is above the threshold, specified by the user

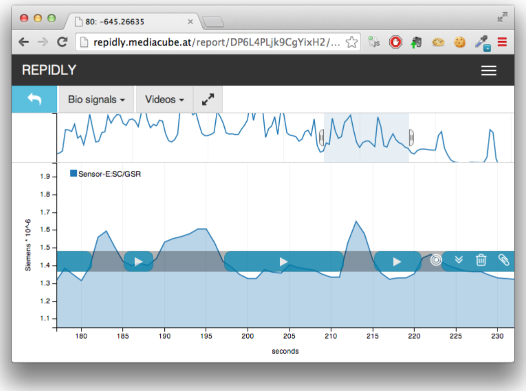

Figure 14: Identification of time intervals, in which the EDA signal is below the threshold, specified by the user

Once a set of time intervals is extracted, it may be pinned to the top of the time series graph for subsequent use, as illustrated in figure 15. These segments will stay in the time series graph, even if the user activates a different biosignal channel, making it possible to visually compare multiple layers of threshold segments, extracted from different biosignal channels. Each threshold segment is symbolized by a play button, in which the button width is mapped to the length of the corresponding time interval. With a click on the play button, the enabled video channels appear ontop of the view and play back the selected time range, as illustrated in figure 16.

This interactive filtering mechanism facilitates the identification of emotionally relevant gameplay situations on the basis of psychophysiological data. To allege an example, figure 17 illustrates a situation, in which a steep increase in the player's EDA level co-occurs with an in-game death event. This use case clarifies, how a targeted inspection of the gameplay video, based on interesting patterns in a biosignal, may accelerate the identification of emotionally relevant in-game events. Without the EDA signal, the death event could only have been reliably detected by watching the video in chronological order, taken that no in-game event logs are available.

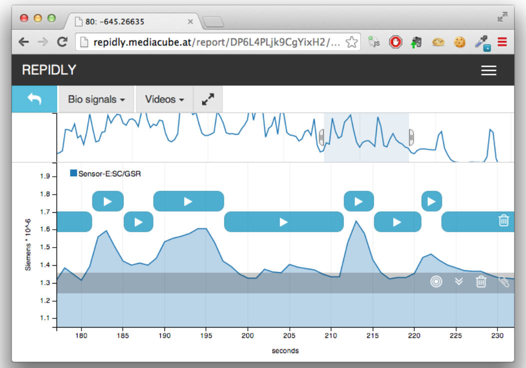

Figure 15: Two layers of threshold segments, pinned to the top of the time series graph

Figure 16: Presentation of a gameplay sequence from two different video perspectives. The sequence was identified by a threshold segment in the EDA channel.

Figure 17: EDA peak associated with an in-game death event

In summary it can be said that although *Repidly*'s visualization concept builds on the BioSt framework, both approaches differ in the process of data exploration. In the BioSt framework, gameplay videos are viewed chronologically by games user researchers and players together in order to insert subjective player comments into the single player's data graph, based on the timestamp in the video. In contrast to that, *Repidly* employes an interactive thresholding mechanism for psychophysiological signals to accelerate the identification of emotionally relevant gameplay situations. The corresponding video excerpts may be further examined and the resulting insights may be put down collaboratively in the form of annotations, which will be covered in the next section.

3.1.3 Annotations

In addition to the interactive drill-down and inspection procedures for biosignal data and videos, *Repidly* enables users to comment on situations in a gameplay session by adding annotations directly into the time series graph. This functionality was implemented to allow users to further specify insights, gained from the corresonding video excerpts, after an interesting biosignal pattern and its corresponding gameplay video segment have been examined. Comments may be inserted into the timeline by clicking in the time series graph. In *Repidly*, each comment is characterized by the following three dimensions:

1. literary (text content)
2. temporal (time position)
3. emotional (positive, neutral or negative sentiment)

Figure 18 presents three comments of different sentiment, placed inside the time series graph. They are symbolized as color-coded stickmen, visualizing negative sentiment with a red fill color, positive sentiment in green and a neutral comment in gray. Figure 19 illustrates the editing interface for comments, which may be opened by clicking on the corresponding stickman symbol in the timeline. It contains three buttons to switch between sentiments as well as a text field, in which multiple users can collaboratively edit the text contents of a comment. As further outlined in section 3.2.3, co-occuring write operations are merged to allow multiple researchers to edit the same text simultaneously.

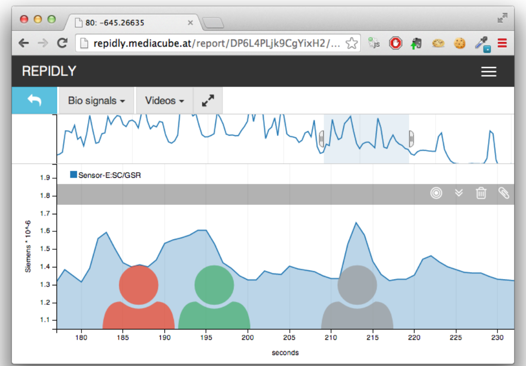

Figure 18: Three comments, placed in the time series graph, with three different sentiments: negative, positive, neutral (from left to right)

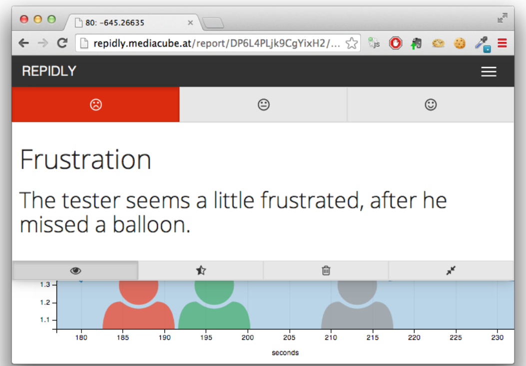

Figure 19: The editing interface for a comment; at the top, three buttons allow to switch between sentiments; underneath, an input field enables collaborative editing of the text contents

3.2 Implementation Details

Repidly was built with several design goals in mind. In this section, it will be explained in more detail how specific functional and software architectural objectives were implemented. In particular, implementation details with regard to the following topics will be covered in the following sections:

1. Portability (section 3.2.1)
2. Extensibility (section 3.2.2)
3. Collaboration features (section 3.2.3)
4. Random Access To Player Data (section 3.2.4)

3.2.1 Portability

In order to make *Repidly* easily portable to all major operating systems, it was developed as a web application, adhering to a set of well-established web standards, consisting of the HyperText Markup Language (HTML) (Navara et al. 2014), Cascading Style Sheets (CSS) (Etemad 2011) and JavaScript (ISO-16262 2011). As web applications are executed in browsers, which are supported on all major operating systems, it does not require further adaptations of *Repidly* to make it operational in these environments. To further support the use of *Repidly* not only on desktop computers, but also on mobile devices, typically featuring relatively small screen sizes, a responsive layout was applied, which automatically adapts its components to the size of the screen, on which the application is presented. Furthermore, using application wrappers, such as Cocoon.js[1] or PhoneGap[2], it is possible to create hybrid mobile applications, which encapsulate a web application in a way that it appears as a native application on the mobile platform. Similarly, web applications may be re-packaged into native desktop applications, using, e.g., Chromium Embedded Framework[3]. Concludingly, the decision to implement *Repidly* as a web application was driven by the rationale that web applications offer a high degree of portability between operating systems, runtime environments and device types.

1. Ludei Inc. "Cocoon.js." Accessed March 24, 2014. `https://www.ludei.com/cocoonjs/`.
2. Adobe Systems Inc. "PhoneGap." Accessed March 24, 2014. `http://phonegap.com/`.
3. Greenblatt, Marshall. "Chromium Embedded Framework." Accessed March 24, 2014. `https://code.google.com/p/chromiumembedded/`.

3.2.2 Extensibility

At its heart, *Repidly* is a tool for the visualization and analysis of player data, which is designed to potentially handle a wide variety of data types. Initially supporting purely static data, ranging from biosignal time series over video recordings to written comments, it is desirable that this set of data types may, in the future, be extended by data types, featuring different characteristics, may it be in terms of data transmission (e.g., streamed data) or data structure (e.g., semi-structured data from in-game event logs). Therefore, *Repidly* is designed to be extensible by other data types, featuring their own characteristics and requiring dedicated database schemas and code logic. This flexibility is ensured by a package structure, in which each supported data type is encapsulated in its own package. Moreover, each data type package is represented by a data type class, which encapsulates server logic, allowing it to dynamically create the required database schemas at runtime.

All data type packages have two interfaces in common:

1. the `DataType` class, which is the base class for all data types
2. the `Data` collection, an abstract database container, keeping one document for every instance of a data type

Figure 20 shows a top-level entity relationship diagram of *Repidly*'s database schema in crow's foot notation. Each entity in the diagram represents a collection in *Repidly*'s Not-only SQL (NoSQL) database MongoDB[4]. In MongoDB, collections are the topmost instances in the database hierarchy. Each collection contains multiple documents. Expressed in terms of relational database systems, collections are the counterparts to tables and documents correspond to table entries.

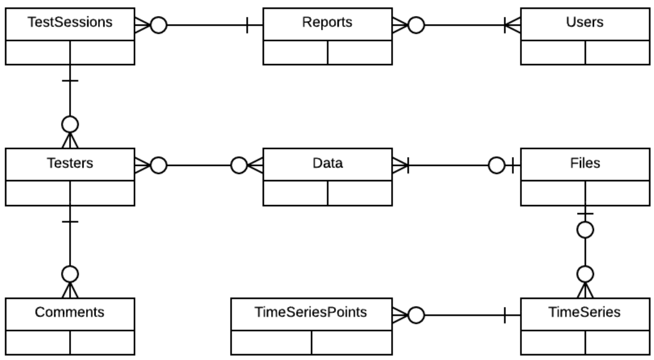

Figure 20: Top-level entity relationship diagram of the database schema.

4. MongoDB Inc. "MongoDB." Accessed April 30, 2014. `http://www.mongodb.org/`.

In figure 20, the `Data` collection is located in the center of the diagram. It is associated with the `Testers` collection in an optional many-to-many relationship. Put another way, each tester may reference an arbitrary number of data entries, while each data entry may be shared among multiple testers. The latter is required in scenarios, where one data entry contains information about more than one tester, e.g., a video recording of several players, engaged in co-located gameplay. Each data entry may optionally reference a file entry, while each file entry may, in turn, be shared among several data entries. The latter relation is required in cases, where several data type instances reference data from one common file, e.g., a compressed folder, containing both video and biosignal data. However, not all data entries originate from file contents, but, as previously mentioned, may as well be continuously transferred to the web server from a streaming source. This is the reason why associations to file entries are kept optional.

However, the entity relationship diagram, shown in figure 20, does not only provide a general overview on *Repidly*'s database structure, but it also exposes an example of how data types are capable of extending the database structure according to their specific requirements. In this particular case, the `TimeSeries` data type, has added the `TimeSeries` and `TimeSeriesPoints` collections to *Repidly*'s datastore in order to store time series, including their associated metadata and point information.

3.2.3 Collaboration Features

Besides being portable and extensible, one of *Repidly*'s design objectives is to allow multiple researchers to work collaboratively on one report. This rather general requirement may be further subdivided into three more specific features, which need to be implemented for a seamless collaboration among multiple application users. The requirements, covered in more detail in the following sections, are user management, write conflict resolution and real-time data synchronization.

User Management

To enable the cooperation between multiple users on a single report, *Repidly* has a built-in user management system, in which users can create their accounts, either using their email address or via social media authorization interfaces. Each report may be associated with a number of users, while each user may in turn be connected to multiple reports, as depicted in figure 20. In particular, users take one of the following roles, when connected to a report: administrator, assistant or viewer.

Only if a user is associated with a report, he/she will be able to access it at all. Viewers have read-only access to a report and can therefore only view, but not modify its contents. In contrast to that, assistants can read and modify a report's actual contents, whereas administrators have full read and write access to all its contents and configurations, being able to add other users to it and decide over their access rights.

Write Conflict Resolution

Once multiple users are associated to a single report, having the ability to modify the same resource simultaneously (e.g., a comment), mechanisms of write conflict resolution become inevitable. They are needed to ensure the integrity of data resources, modified by more than one user at a time. To take an example, if two users attempt to edit the text of a comment virtually at the same time, it is not sufficient to persist only the latter of the two write operations by overriding the first one. Instead, for collaborative text editing in particular, it is important to merge simultaneously occuring write operations by applying algorithms for write conflict resolution, such as operational transformation (OT) (cf. Shao et al. 2011). For *Repidly*, the open-source JavaScript OT library ShareJS[5] was employed for this purpose.

Real-time Data Synchronization

In addition to an effective way to merge write operations, another challenge in the creation of a collaborative web application lies in the reflection of data changes to all application users in real time. Traditionally, web applications rely on a sequence of network requests, based on the HyperText Transfer Protocol (Fielding et al. 1999), each delivering resources, such as HTML files, style sheets or JavaScript files, from the web server to the requesting client. In this traditional scenario, a new HyperText Transfer Protocol (HTTP) request is initiated, whenever a user of a website clicks on a hyperlink. After delivery of the content, the Transmission Control Protocol (TCP) connection, used for data transmission, will be terminated again, making a persistent, two-sided data connection between the server and client impossible.

With the standardization of the WebSocket API (Hickson 2012) and its implementation in all major browsers, however, this type of persistent data connection between web clients and servers has become feasible, leading to the development of real-time data synchronization frameworks for web application development. For the implementation of *Repidly*, the open source project Meteor[6] was employed as the core development framework, enabling real-time data synchronization among multiple clients.

Figure 21 depicts the general architecture of an interactive web application, showing the server architecture on the left and the client architecture on the right side. In this setup, server and client have a separate code base and communicate solely via API endpoints, provided by the server. Only the server owns a datastore, from which it may transfer data to the client, if requested. The client view is defined in HTML and CSS and its document object model (DOM) may be manipulated, using JavaScript commands.

5. Gentle, Joseph. "ShareJS." Accessed March 24, 2014. `http://sharejs.org/`.
6. Meteor User Group. "Meteor." Accessed March 24, 2014. `http://meteor.com`.

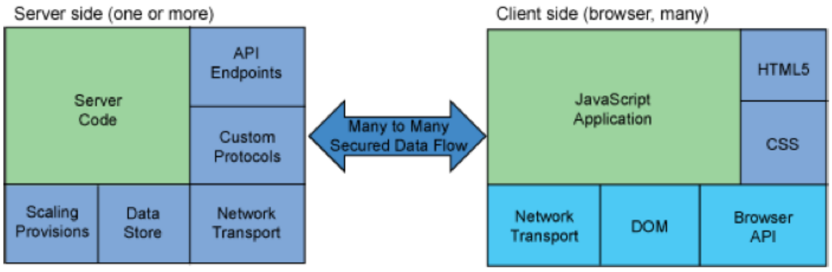

Figure 21: Rich-client interactive web application architecture (reprinted from Sing 2013, p. 24)

Figure 22 shows the client-server setup of a Meteor application. In this setup, the server code runs ontop of Node.js[7], an event-driven, non-blocking JavaScript runtime environment. Both server and client code are authored in JavaScript, allowing for a shared code base between the two. All persistent data is stored in a MongoDB database on the server and may be subscribed by the client, if the server was programmed to publish the data to the respective client. If a client subscribes to a specific data set, the data will be transferred to the client and changes to it will be reflected in real time via a persistent WebSocket connection. For these dynamic data updates, a custom protocol, called dynamic data protocol (DDP), was created for Meteor. The Meteor client receives all page layouts from the server on its initial HTTP request and stores them locally. After that, a persistent DDP connection between the server and the client is established and used to repeatedly propagate data changes over the network.

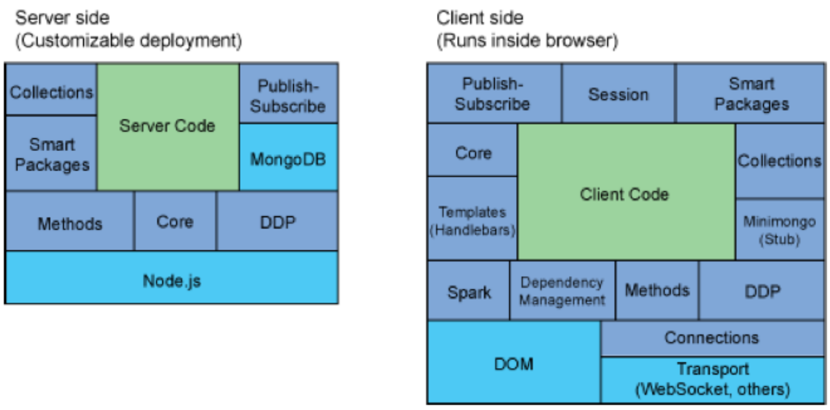

Figure 22: Meteor internal components (reprinted from Sing 2013, p. 25)

Concludingly, the Meteor web server actively distributes data changes to all subscribing clients in real time. In combination with user management and write conflict resolution, real-time data synchronization thus forms a solid basis for a functional collaborative web application, in which data modifications are instantly reflected in all client views.

7. Joyent Inc. "node.js." Accessed April 30, 2014. `http://nodejs.org/`.

3.2.4 Random Access To Player Data

In *Repidly*, the main bottleneck of server payload is the handling of large data sets. To take an example, a three-channel biosignal recording of a five minutes long gameplay session, sampled at 2048 Hertz (Hz), results in a file with a size of 20 megabytes (MBs), containing approximately 1800000 data points. The amount of data to be handled per file scales up linearly with the duration of gameplay sessions and the sampling rate of the recordings respectively. Therefore, to allow games user researchers to collaboratively analyze player data through a web interface, it is crucial that data are stored in a way that researchers are able to randomly access any section of the data with a constantly short waiting time. To take an example, it would be fatal, if a researcher had to download the entire data of a biosignal file everytime he/she intended to view just an excerpt of the recording, because 1800000 data points would be required for generating the excerpt of the graph. The same consideration holds true for the playback of large video files, where it is crucial for researchers to be able to watch just parts of a long video recording without having to download the entire video file. In this section, it will be explained in detail, how random access to video and time series data was implemented in *Repidly*.

HTTP pseudo-streaming for videos

HTTP pseudo-streaming is a method to allow random access to video files, located on a web server. As the name suggests, it is based on the HyperText Transfer Protocol. On the server side, it requires a specialized interface, which takes byte range requests from the client and performs a targeted read operation on the video file to read and deliver only the byte range, requested by the client. On the client side, a HTTP pseudo-streaming compatible video player must be employed. For the development of *Repidly*, the open-source video player Flowlayer[8] was deployed. That way, users of *Repidly* are enabled to navigate to an arbitrary section of a video without having to download the skipped part. The main benefit of HTTP pseudo-streaming over other streaming techniques like the real-time transport protocol (RTP) (Schulzrinne et al. 1996) or real-time messaging protocol (RTMP) (Parmar and Thornburgh 2012) is the fact that it can be implemented on a regular HTTP-based web server, minimizing the risk of a request being blocked by a firewall.

8. Flowplayer Ltd. “Flowplayer.” Accessed April 30, 2014. `http://flowplayer.org/`.

Level-of-detail implementation for time series data

As previously mentioned, random access to arbitrary data segments in constant time is not only required for video-, but also for time series data. As there is no plug-and-play solution available to solve this problem on a Node.js- and MongoDB-based web server, a custom level-of-detail (LOD) implementation for time series data was implemented for *Repidly*. The basic idea behind the LOD implementation is that the server will only transmit those data points, needed to generate the excerpt of the time series graph, the client is currently viewing. Furthermore, the server will send the data in the most suitable resolution, depending on the width of the screen area w, used to visualize the time series segment. This approach makes it possible to cap the maximum number p of data points, being transferred over the network. Correspondingly, using $p > w$ data points for the visualization of a time series segment would mean trying to render more data points than there are pixels available for presentation.

Figure 23 shows two different time range selections of the same time series. The broader selection (top half) shows the selected segment at a lower resolution than the more narrow selection (bottom half). In other words: the deeper the user zooms into the time series, the more detailed the selected segment will be presented.

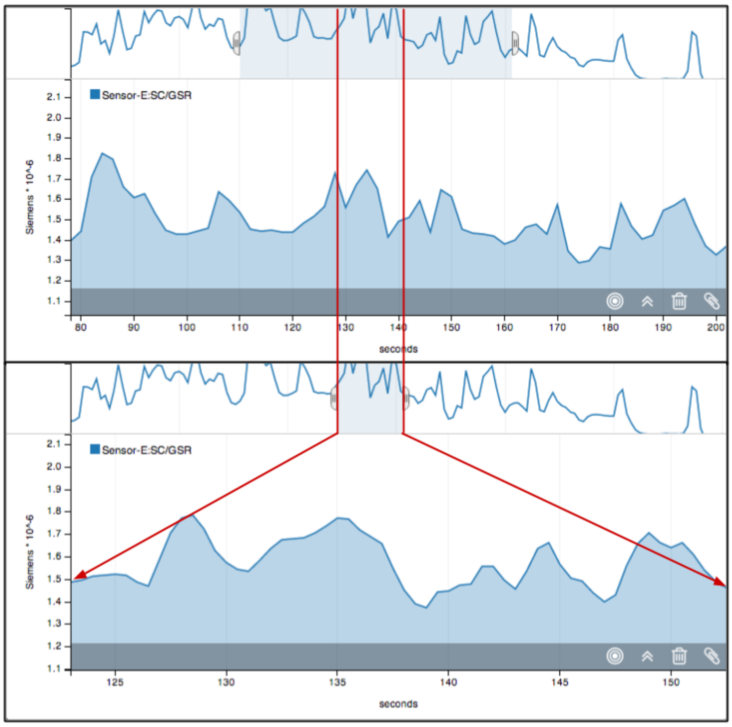

Figure 23: Level-of-detail implementation for time series data. If a relatively large segment of the time series is selected (top half), it will be presented at a relatively low resolution. If the selection is drilled down to a smaller segment (bottom half), the time series data will be rendered with a higher degree of detail.

For this mechanism to work, however, the time series data needs to be available at multiple resolutions, which is achieved by a multi-level aggregation process, based on the computation of averages over multiple data points. Each resolution may be identified by an aggregation level l, indicating the length of time series segments (in seconds) that were averaged to compute the resolution. To take an example, each data point of a resolution with $l = 2$ is the average of a two seconds long interval (with the exception of the last data point in the resolution, which might result from a shorter time interval).

Figure 24 illustrates how three different resolutions (with aggregation levels 2, 5 and 10) are computed for a given time series of length 18 seconds. Each resolution with an aggregation level l_1 is computed from a higher resolution with aggregation level l_0, where l_0 is the largest divisor of l_1 among all available resolutions. To give an example, in the aggregation procedure, presented in figure 24, the resolution with $l_1 = 10$ is computed directly from the next higher resolution with $l_0 = 5$, because 5 is the largest divisor of 10 among all available aggregation levels (1, 2 and 5). This optimization procedure significantly reduces the number of computational operations, needed to perform a stepwise multi-level aggregation of time series data.

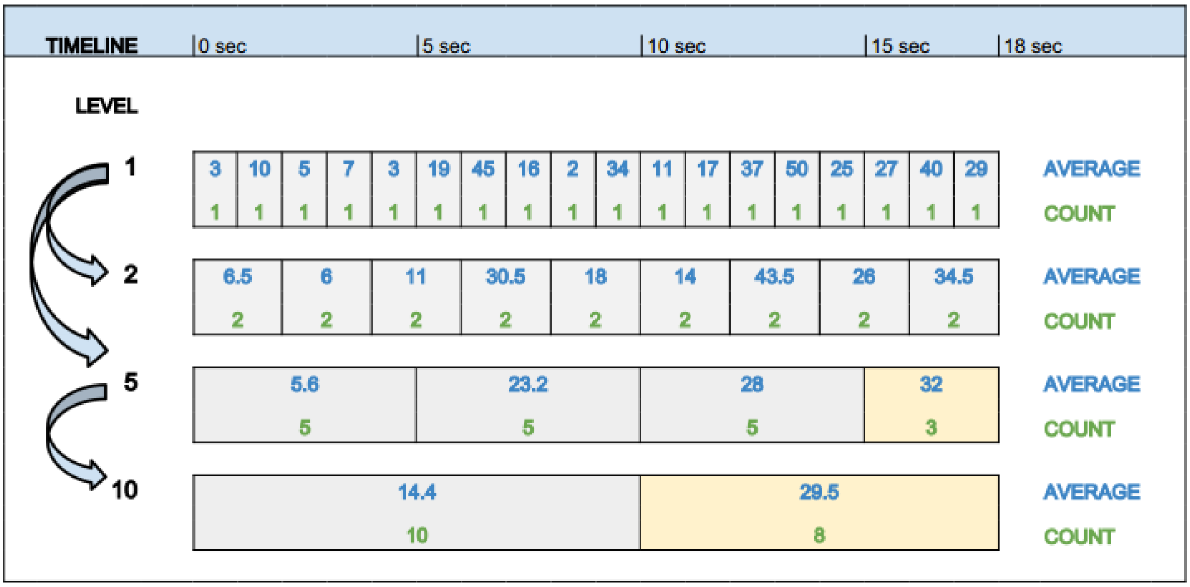

Figure 24: Schema of a multi-level time series aggregation, based on an 18-seconds-long time series with an initial aggregation level of 1, i.e., a sampling rate of 1 Hz (upper grey box array). The subsequent aggregations for levels 2, 5 and 10 are shown in the three box arrays underneath. Aggregation levels 2 and 5 are computed from aggregation level 1, whereas aggregation level 10 is derived from level 5. The average values are color-coded in blue (AVERAGE), whereas the green integers (COUNT) indicate the number of values from the initial aggregation level that contribute to the average value. The yellow boxes indicate aggregation steps, in which the number of averaged values (green) is smaller than the corresponding aggregation level (3 < 5 and 8 < 10).

In *Repidly*, the multi-level aggregation takes place after the raw time series data has been inserted into the database. MongoDB offers two different built-in interfaces to perform aggregation on data[9]: (1) the aggregation pipeline, which performs better for simple aggregations, but is constrained by more usage restrictions and (2) the MapReduce framework, which is more flexible, but generally less performant (see Dean and Ghemawat 2008 for an introduction to the MapReduce paradigm).

Both interfaces were explored during the development process of *Repidly*. It could be found that the MapReduce-based implementation results in slightly more readable code, because it allows a functional syntax, based on regular JavaScript instructions. In contrast to that, the aggregation pipeline requires the entire aggregation process to be defined in Javascript object notation (JSON)[10], resulting in a deeply nested, complex object. For the complete implementation of both approaches, the reader is referred to appendix section A (see listing 1 for an implementation, based on the MapReduce framework, and listing 2 for an implementation, based on the aggregation pipeline).

As outlined in this section, HTTP pseudostreaming and a custom LOD implementation for time series data were employed in *Repidly* to provide random access to video recordings and biosignal time series respectively. It was explained in detail that the time series LOD implementation requires data points to be post-processed on the server side in order to make time series available at different resolutions. The processing of large data sets, however, can be a computationally expensive task. Therefore, a cluster-based server architecture will be proposed in the next section, presenting an option to keep *Repidly*'s data processing capabilities scalable without affecting its functionality for real-time data synchronization.

9. MongoDB Inc. "MongoDB Aggregation Concepts." Accessed March 26, 2014. `http://docs.mongodb.org/manual/core/aggregation/`.

10. Ecma International. 2013. "Standard ECMA-404 The JSON Data Interchange Format." `http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf`.

3.3 Server Architecture for Extended Scalability

In the context of server applications, the term scalability refers to the ability of adding resources to a server application in order to handle computationally more expensive or more frequent network requests (cf. Garcia et al. 2008). As stated in section 3.2.4, the major server-side payload for *Repidly* lies in the processing of large data sets, originating from a relatively small number of network requests. Sing 2013 proposes an application infrastructure for real-time big-data web applications, based on Meteor. It involves an interface between the document-oriented datastore MongoDB and Hadoop[11], an open-source framework for cluster-based, large-scale storage and processing of scientific data (cf. Dede et al. 2013).

Figure 25 depicts the client-server architecture, suggested by Sing 2013, in which a Hadoop cluster works ontop of a MongoDB database instance. Triggered by task queues from the Meteor web server, it extracts data from MongoDB, processes it and writes it back into the MongoDB datastore. The resulting data changes are then propagated to an interactive dashboard in real time, relying on Meteor's data synchronization capabilities.

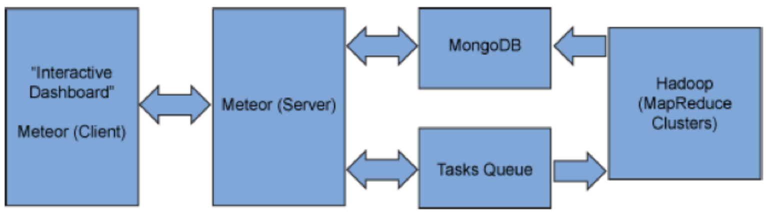

Figure 25: Architecture of a Meteor dashboard application for big data (reprinted from Sing 2013, p. 25)

Dede et al. 2013 contribute a performance evaluation of a MongoDB- and Hadoop-based platform for scientific data analysis. In their tests, the runtime performance in MapReduce jobs with varying input sizes is compared between MongoDB, Hadoop and a combination of both. Figure 26 illustrates the high-level strategy for communication between MongoDB instances and Hadoop clusters, as used in the study. It also depicts Hadoop's strategy for parallelizing data processing tasks, based on the MapReduce approach (Dean and Ghemawat 2008). In this setup, multiple Hadoop server nodes, called mappers, read data splits from MongoDB, using the open-source connector module mongo-hadoop[12], and process the data chunks in parallel. When all mappers have completed their data transformation tasks, a different Hadoop server node, called reducer, joins the resulting data together and streams it back into the MongoDB datastore.

11. The Apache Software Foundation. "Apache Hadoop." Accessed April 30, 2014. `http://hadoop.apache.org/`.
12. MongoDB Inc. "MongoDB Connector for Hadoop." Accessed April 30, 2014. `http://docs.mongodb.org/ecosystem/tools/hadoop/`.

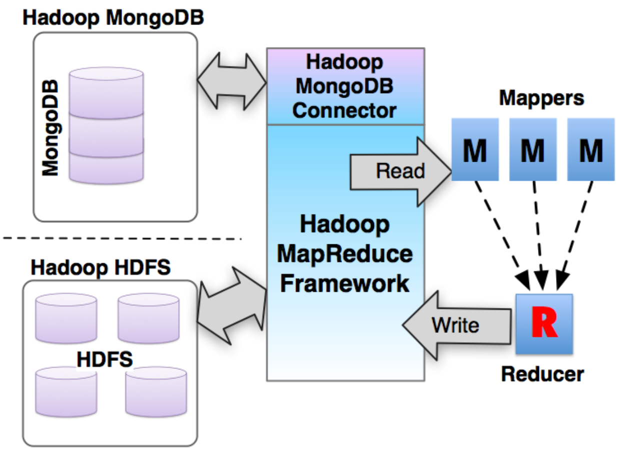

Figure 26: The high-level architecture of mongo-hadoop. Multiple mappers read the input splits from either MongoDB (via the Hadoop/MongoDB connector) or HDFS. The intermediate output is collected in one reducer, which writes the results back to MongoDB or HDFS (reprinted from Dede et al. 2013, p. 2).

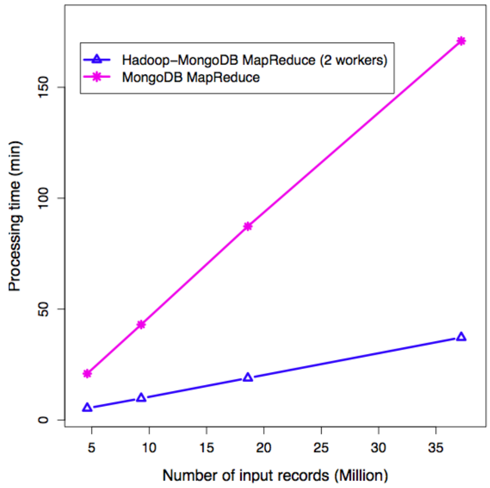

Figure 27: Native MongoDB MapReduce (MR) versus mongo-hadoop MR. Experiment used 1 MongoDB server and, for mongo-hadoop, 2 Hadoop workers (reprinted from Dede et al. 2013, p. 5).

Figure 27 shows the graph of a performance comparison between MongoDB's built-in MapReduce framework and Hadoop's MapReduce framework, when operating ontop of MongoDB. The graph indicates that two Hadoop workers, connected to a MongoDB instance, perform around six times better than the MongoDB instance alone, independently of the input size. Concludingly, an architecture, in which the Meteor, MongoDB and Hadoop operate in conjunction, entails the potential to significantly scale up *Repidly*'s data processing capabilities without compromising its functionality for real-time data synchronization.

4 Discussion and Contribution

In the previous sections, it was outlined in detail, how the applicability of psychophysiological data as real-time GUR metrics as well as the visualization of player data have been investigated from different perspectives. However, the largest amount of research in the field of player data visualization is restricted to behavioral metrics, acquired by telemetry. Only few studies discuss the visualization of psychophysiological player data as a way to facilitate the evaluation of video games. The BioSt approach is one of the few examples, in which biometric and behavioral player data as well as gameplay video recordings and subjective annotations are combined into a coherent visualization of gameplay experience. However, no implementation of these visualization principles as a collaborative software application for games user researcher has been proposed in the literature so far.

Therefore, the core contribution of this thesis to the fields of GUR and information visualization is the design and implementation of *Repidly*, a software application, in which games user researchers may visually combine biosignals and gameplay videos to annotate gameplay sessions in a collaborative way. Meaningful patterns in psychophysiological data may be filtered in an interactive manner to accelerate the process of identifying gameplay situations, which are relevant for improvements to the game design. Consequently, the presentation of the BioSt approach in section 2.2.2 as well as the visualization concept and functionality of *Repidly* offer comprehensive answers to the question, how psychophysiological signals and gameplay videos can be visually combined to facilitate the process of video game evaluation (research question 1).

In section 3.2 of this thesis, the technical implementation of *Repidly* as a collaborative web application was described in detail, providing an answer to the question, how existing web technologies may be extended to create a online, collaborative GUR tool. In particular, it has been outlined in detail, how video and biosignal data may be stored, processed and made available online. HTTP pseudostreaming and multi-level aggregation of time series were highlighted as means to allow random access to large video and time series data respectively. Furthermore, it was explained, how the collaboration among mutliple users on player data was made possible by means of user management, write conflict resolution and real-time data synchronization. All these techniques were implemented on the basis of the web application development framework Meteor, backed by the JavaScript runtime environment Node.js and the NoSQL datastore MongoDB.

Finally, section 3.3 further addressed the technical challenge of scaling up *Repidly*'s data processing capabilities without compromising its functionality for real-time data synchronization. In particular, Hadoop, a cluster-based system for the large-scale analysis of scientific data, was introduced as a component to process data, originating from a MongoDB datastore, in a scalable fashion. It was shown, how an optional extended server architecture, based on a combination of Meteor, MongoDB and Hadoop, may be used to ensure sufficient data processing capabilities without forfeiting the data synchronization features, needed to enable a seamless cooperation among mutliple users. Summing up, sections 3.2 and 3.3 provided an extensive answer to the question, how existing web technologies can be extended to create a software application for games user researchers, that works online in a large, collaborative setting (research question 2).

Based on the specific characteristics of psychophysiological signals and the way, *Repidly* was implemented as a collaborative web application, different GUR scenarios will be described in the following, in which the use of *Repidly* is of particular benefit for games user researchers.

Tracking emotional valence in fast-paced games

In fast-paced games, such as FPSs, it is a difficult endeavor to associate emotional responses to specific in-game events, using post-hoc questionnaires or interviews. One of the reasons is that players might not be able to recall specific events and changes in their subjective states, after the gameplay session has ended. Instead, players might tend to report a diffuse experience, originating from a number of different in-game events. This makes it difficult to associate fast-paced in-game events with specific emotional reactions.

Electrophysiological techniques, such as EMG or EDA, however, are capable of registering instant psychophysiological reactions with sampling rates of around 256 - 2048 Hz, lying above the maximum frame rates of modern gaming computers (60 - 150 Hz). The temporal precision of these measures is therefore high enough to compensate for rapid sequences of potentially meaningful in-game events and thus presents an adequate way to infer emotional reactions of players, engaged in fast-paced games (cf. Drachen et al. 2010).

Therefore, the *Repidly* prototype, presented in this thesis, presents a strong contribution for the evaluation of emotions, aroused in fast-paced games, by offering games user researchers a possibility to cooperatively annotate video recordings, based on the visualization of the abovementioned psychophysiological signals.

Tracking the effects of subtle audiovisual ambience

In many game genres, ambient audiovisual effects are used to convey mostly subtle atmospheres to the player in reaction to in-game events. Examples are the crowd ambience in sports games, the ambient sound of sirenes in open world city crime genres or the audiovisual atmosphere in horror games. As Bridgett 2013 points out, interactive ambient sound can have a “powerful subliminal effect on the player and raises or diffuses tension in a very subtle way”. Although these kinds of subtextual effects might not be consciously perceived by the player in many situations, they might well contribute to the overall play experience, when the timing of the effect is appropriate in reaction to the player’s actions.

The use of sensitive recording devices to measure emotionally relevant biosignals, such as EDA and facial EMG, presents a way to register even fine-grained psychophysiological reactions to subtle in-game stimuli, such as ambient audiovisual effects. Therefore, games user researchers might use *Repidly* to identify subliminal emotional responses, not consciously perceived by the player, in order to analyze the effectivity of subtle audiovisual effects in digital games.

Evaluating emotion-centered games

Some game development studios make emotions the central element in the process of game design. Vander Caballero, designer of *Papo & Yo* at *Minority Media Inc.*, expresses this idea in his intention to "take someone on an emotional journey" (Graft 2014). In scenarios, where emotions are the key element of the game design and development process, software tools like the one presented in this thesis might offer ways to analyze diffuse emotional components on the basis of biosignals, unobtrusively tracked in real time. When combined with qualitative data from player interviews and questionnaires, a tool like *Repidly* might therefore provide a powerful instrument to facilitate the evaluation of emotion-centered games in particular.

Cooperation between geographically remote teams

In an increasingly globalized world, the cooperation between geographically remote teams has become a part of the regular work life of many individuals. In the context of the video games industry, it is especially the evaluation of game prototypes that is often outsourced by small or medium-sized companies, lacking the resources for an in-house user research department. Working in remote teams, however, often complicates communication processes and the exchange of work material such as documents or, in the case of GUR, player data and analytic reports. Against this background, *Repidly* fills the gap of a collaborative software application for the remote cooperation between games user researchers and development teams. Consequently, it presents a meaningful contribution for the GUR community by promoting the efficiency of games user researchers, collaborating over a geographic distance.

Summing up, this thesis contributes a powerful instrument for games user researchers, especially against the background of evaluating emotion-centered and fast-paced games as well as analyzing the effects of subtle audiovisual ambience. Moreover, it presents a novel way for games user researchers to cooperate over a geographic distance by proposing a software application for video game evaluation, working online in a large, collaborative setting. Limitations of the concept and its current implementation will be addressed in the following section.

Limitations

In the current implementation, *Repidly* provides interactive visualization and filtering techniques for single-channel biosignals, in which the amplitude of the signal allows for direct inferences about components of the underlying emotional state of the player (e.g., EDA and facial EMG). As stated in section 2.1.2, other biosignals, such as EEG, exhibit more complex structures, requiring more sophisticated techniques for data analysis and representation. The *Repidly* prototype, however, lacks meaningful visualization concepts for complex, multi-channel biosignals. Sticking to the example of EEG, this shortcoming could be addressed by introducing frequency analysis as an additional processing step after the upload of an EEG file. A single EEG channel could be visualized, e.g., as a spectrogram, aligned with the timeline in the data visualization view. Multiple EEG channels may be grouped by electrode location (e.g., frontal, parietal, occipital), allowing for aggregation of group data, e.g., by computing and visualizing the average power of a given frequency band at a specific point in time for all frontal electrodes.

Another limitation of the current prototype is that users cannot logically combine multiple filters for different time series channels. A common use case for this functionality would be to identify situations, in which the player exhibits high values in both emotional arousal (i.e., EDA) and valence (i.e., EMG of the ZM muscle). Instead, layers of threshold segments need to be pinned to the top of the time series graph (cf. figure 15 in section 3.1.2) in order to be compared on a merely visual basis. This limitation could be alleviated by introducing Boolean operations to be applied to multiple layers of extracted time intervals. Using Boolean operations, this task could be achieved by first filtering the appropriate EDA and facial EMG channels for above-threshold time segments and, in a second step, combining both layers, using a union operation.

Moreover, the prototype does not allow researchers to associate players with participant groups and player data with experimental conditions, as these concepts do not exist in *Repidly*'s data model, though required in many experimental settings, such as between-groups or within-subjects designs. Adding these concepts to *Repidly*'s data model would allow researchers to analyze the effect of certain game design elements not only correlationally, but also experimentally. In addition, there is no built-in functionality available to export analytic results as static reports. In future versions of the prototype, these reports might be created semi-automatically on the basis of annotations, customizable data views and static images from video excerpts.

Finally, it should be noted that the current prototype does not yet offer a way to automatically align different data recordings. Instead, the timeline visualization is based on the assumption that all data recordings, represented in the view, start exactly at the same point of time. The problem of temporal alignment could be addressed by unique physical markers (e.g., a flash light in a video or a voltage peak in a biosignal), simultaneously inserted into multiple recordings in order to indicate the starting point of a gameplay session. These physical markers might be computationally detected and used for data alignment.

5 Conclusion

Over the last few years, biosignals have been examined as metrics in games user research due to their applicability as unobtrusive, real-time and high-precision measures, used to quantify emotional components of gameplay experience, such as arousal (through, e.g., EDA) and valence (through, e.g., facial EMG). A growing body of academic research addresses the questions, how biometrics can be effectively employed as measures in the evaluation of video games (cf. Mandryk 2008, Nacke and Lindley 2008, Drachen et al. 2010, Kivikangas et al. 2010, Nacke, Grimshaw, and Lindley 2010, Nacke 2011 and Wehbe et al. 2013) and how the contribution of biometrics to toolset of games user researchers may be quantified (cf. Mirza-Babaei et al. 2011). However, not only in academic GUR settings, biometrics have become an increasingly relevant topic. Also representatives of large industrial video game studios, such as *Valve Corporation* (Ambinder 2011), *Electronic Arts, Inc.* (Zammitto 2011) and *Ubisoft Entertainment S.A.* (Chalfoun 2013) confirm the applicability of psychophysiological signals as evaluation metrics for digital games.

Despite the increasing popularity of biometrics in the GUR context, only few studies have addressed the challenge of integrating intuitively accessible visualizations of these metrics in software tools for games user researchers. The majority of research in the field of player data visualization places special emphasis on behavioral data, derived from in-game events. Most approaches focus on the visualization of spatiotemporal data in the form of heatmaps or motion trajectories (cf. Drachen 2013 and Drachen 2014). In other studies, gameplay is represented as an abstract finite state machine and visualized as a node-link diagram (e.g., in Wallner and Kriglstein 2014). The BioSt approach, proposed by Mirza-Babaei et al. 2013b, forms an exception in that it combines biosignal data from EDA and facial EMG recordings, together with gameplay videos, behavioral player data and player annotations to form a coherent visualization of gameplay experience.

Even though the BioSt approach addresses the issue of visualizing biometrics in combination with other types of player data, it leaves open the question, how the process of visualizing and analyzing biometric data may be implemented in a large, collaborative setting, as required, e.g., by big industrial studios. In response to this shortcoming, the *Repidly* prototype was presented in this thesis, a web-based, collaborative GUR tool, designed to facilitate the annotation of gameplay videos, based on an interactive visualization of biosignal time series. The visualization concepts, applied in *Repidly*, were inspired by the BioSt approach (Mirza-Babaei et al. 2013b). In the course of this thesis, it was elaborated on *Repidly*'s software architecture, designed for a high degree of extensibility with regard to data types and enabling random access to arbitrary segments of even large-sized data sets. Finally, the contribution of this work with particular regard to emotion-centered and fast-paced games as well as the evaluation of subtle audiovisual ambience effects and the cooperation among geographically remote GUR teams was highlighted on the basis of multiple usage scenarios.

Future Work

In future studies, the current version of the *Repidly* prototype needs to be tested in a usability study with GUR experts and the effectivity of different biosignals in the context of *Repidly* needs to be analyzed. In addition, the application's data processing performance and scalability need to be quantified in future work on the topic.

The current implementation supports visualizations only for single-channel biosignals, such as EDA or facial EMG. Therefore, the challenge of visualizing complex multi-channel signals, such as EEG, to make relevant patterns intuitively accessible for games user researchers might be addressed in future enhancements to the approach. Apart from that, it would also be a meaningful contribution to incorporate other types of data, such as questionnaires or behavioral player data from in-game event logs, into the collaborative evaluation process of *Repidly*. This brings up the question, how behavioral and psychophysiological player data may be combined to construct a meaningful visualization of gameplay experience, that is considered helpful by games user researchers.

Furthermore, as the process of manually inspecting extensive data sets is a tedious task, the automatic detection of frequent and surprising patterns in biosignals becomes a desirable feature for GUR tools, especially when long gameplay sessions need to be analyzed. In the field of time series data mining, some research groups have looked at, e.g., the extraction of interpretable patterns of muscular activation (Mörchen, Ultsch, and Hoos 2005) or the classification of emotional biosignal patterns, evoked when viewing affective pictures (Frantzidis et al. 2010). Future research may build on these and other studies to explore the potentials of biosignal data mining techniques for *Repidly* in order to further extend the toolset and optimize the workflows of games user researchers.

List of Figures

List of listings

References

Ambinder, Mike. 2011. *Biofeedback in Gameplay: How Valve Measures Physiology to Enhance Gaming Experience.* Presentation at Game Developers Conference (GDC) 2011. `http://www.gdcvault.com/play/1014734/Biofeedback-in-Gameplay-How-Valve`.

Birbaumer, Niels, and Robert F. Schmidt. 2005. *Biologische Psychologie.* Springer-Lehrbuch. Springer. ISBN: 9783540254607.

Bridgett, Rob. 2013. *Why ambient sound matters to your game.* `http://www.gamasutra.com/view/news/200150/Why\_ambient\_sound\_matters\_to\_your\_game.php`.

Cacioppo, John T., Louis G. Tassinary, and Gary Berntson. 2007. *Handbook of Psychophysiology.* Cambridge University Press. ISBN: 9780521844710.

Chalfoun, Pierre. 2013. *Biometrics In Games For Actionable Results.* Presentation at Montréal International Game Summit (MIGS) 2013.

Dean, Jeffrey, and Sanjay Ghemawat. 2008. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM* (New York, NY, USA) 51 (1): 107–113. ISSN: 0001-0782, doi:`10.1145/1327452.1327492`, `http://doi.acm.org/10.1145/1327452.1327492`.

Dede, Elif, Madhusudhan Govindaraju, Daniel Gunter, Richard Shane Canon, and Lavanya Ramakrishnan. 2013. Performance evaluation of a MongoDB and hadoop platform for scientific data analysis. *Proceedings of the 4th ACM workshop on Scientific cloud computing - Science Cloud '13* (New York, New York, USA):13. doi:`10.1145/2465848.2465849`, `http://dl.acm.org/citation.cfm?doid=2465848.2465849`.

Drachen, Anders. 2013. Spatial Game Analytics. Chap. 17 in *Game analytics: maximizing the value of player data,* 365–402. London: Springer-Verlag. ISBN: 978-1-4471-4768-8, doi:`10.1007/978-1-4471-4769-5`.

———. 2014. *Visualizing Dynamic Behavior Flow.* `http://www.gamasutra.com/blogs/AndersDrachen/20140410/215252/Visualizing\_Dynamic\_Behavior\_Flow.php` (accessed Apr. 19, 2014).

Drachen, Anders, Lennart E. Nacke, Georgios Yannakakis, and Anja L. Pedersen. 2010. Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. In *Proceedings of the 5th acm siggraph symposium on video games,* 49–54. ACM.

Etemad, Elika. 2011. *Cascading Style Sheets (CSS) Snapshot 2010.* {W3C} Note. W3C.

Fielding, Roy T., Jim Gettys, Jeffrey Mogul, Henrik Frystyk, Larry Masinter, Paul Leach, and Tim Berners-Lee. 1999. *Hypertext Transfer Protocol – HTTP/1.1.*

Frantzidis, Christos A., Charalampos Bratsas, Manousos A. Klados, Evdokimos Konstantinidis, Chrysa D. Lithari, Ana B. Vivas, Christos L. Papadelis, Eleni Kaldoudi, Costas Pappas, and Panagiotis D Bamidis. 2010. On the classification of emotional biosignals evoked while viewing affective pictures: an integrated data-mining-based approach for healthcare applications. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* 14, no. 2 (Mar.): 309–18. ISSN: 1558-0032, doi:10.1109/TITB.2009.2038481, http://www.ncbi.nlm.nih.gov/pubmed/20064762.

Garcia, Daniel F., Rodrigo Garcia, Joaquín Entrialgo, Javier Garcia, and Manuel Garcia. 2008. Experimental Evaluation of Horizontal and Vertical Scalability of Cluster-based Application Servers for Transactional Workloads. In *Proceedings of the 8th conference on applied informatics and communications,* 29–34. AIC'08. Stevens Point, Wisconsin, USA: World Scientific / Engineering Academy / Society (WSEAS). ISBN: 978-960-6766-94-7, http://dl.acm.org/citation.cfm?id=1503829.1503833.

Graft, Kris. 2014. *Designing for empathy, with Papo & Yo dev Minority Media.* http://gamasutra.com/view/news/215340/Designing_for_empathy_with_Papo__Yo_dev_Minority_Media.php (accessed Apr. 19, 2014).

Hakvoort, Gido, Hayrettin Gürkök, Danny Plass-Oude Bos, Michel Obbink, and Mannes Poel. 2011. Measuring immersion and affect in a brain-computer interface game. In *Proceedings of the 13th ifip tc 13 international conference on human-computer interaction - volume part i,* 115–128. INTERACT'11. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-642-23773-7, http://dl.acm.org/citation.cfm?id=2042053.2042069.

Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, and Verplank. 1992. *ACM SIGCHI curricula for human-computer interaction.* Technical report. New York, NY, USA.

Hickson, Ian. 2012. *The WebSocket API.* Candidate Recommendation. W3C, Sept.

ISO-13407. 1999. *Human-centred design processes for interactive systems.* Geneva, Switzerland: International Organization of Standardization (ISO).

ISO-16262. 2011. *Information technology - Programming languages, their environments and system software interfaces - ECMAScript language specification.* Norm. Geneva, Switzerland.

Kivikangas, J. Matias, Inger Ekman, Guillaume Chanel, Simo Järvelä, Ben Cowley, Pentti Henttonen, and Niklas Ravaja. 2010. Review on psychophysiological methods in game research. In *Proc. of 1st nordic digra, digra.*

Klinke, Rainer, and Christian Bauer. 2005. *Physiologie.* Thieme. ISBN: 9783137960058.

Luck, Stephen J. 2005. *An introduction to the event-related potential technique.* Cognitive neuroscience. MIT Press. ISBN: 9780262621960.

Mandryk, Regan. 2008. Physiological Measures for Game Evaluation. In *Game usability: advice from the experts for advancing the player experience.*

Mirza-Babaei, Pejman, Sebastian Long, Emma Foley, and Graham McAllister. 2011. Understanding the contribution of biometrics to games user research. In *Proc. digra.* Citeseer.

Mirza-Babaei, Pejman, Veronica Zammitto, Jörg Niesenhaus, Mirweis Sangin, and Lennart E. Nacke. 2013a. Games User Research: Practice, Methods, and Applications. In *Chi '13 extended abstracts on human factors in computing systems,* 3219–3222. CHI EA '13. New York, NY, USA: ACM. ISBN: 978-1-4503-1952-2, doi:10.1145/2468356.2479651, http://doi.acm.org/10.1145/2468356.2479651.

Mirza-Babaei, Pejman, Lennart E. Nacke, John Gregory, Nick Collins, and Geraldine Fitzpatrick. 2013b. How Does It Play Better?: Exploring User Testing and Biometric Storyboards in Games User Research. In *Proceedings of the sigchi conference on human factors in computing systems,* 1499–1508. CHI '13. New York, NY, USA: ACM. ISBN: 978-1-4503-1899-0, doi:10.1145/2470654.2466200, http://doi.acm.org/10.1145/2470654.2466200.

Mörchen, Fabian, Alfred Ultsch, and Olaf Hoos. 2005. Extracting Interpretable Muscle Activation Patterns with Time Series Knowledge Mining. *Int. J. Know.-Based Intell. Eng. Syst.* (Amsterdam, The Netherlands, The Netherlands) 9, no. 3 (Aug.): 197–208. ISSN: 1327-2314, http://dl.acm.org/citation.cfm?id=1233864.1233868.

Nacke, Lennart E. 2011. Directions in Physiological Game Evaluation and Interaction. In *In chi 2011 bbi workshop proceedings.*

———. 2013. An Introduction to Physiological Player Metrics for Evaluating Games. Chap. 26 in *Game analytics: maximizing the value of player data,* 585–620. London: Springer-Verlag. ISBN: 978-1-4471-4768-8, doi:10.1007/978-1-4471-4769-5.

Nacke, Lennart E., Mark N. Grimshaw, and Craig A. Lindley. 2010. More Than a Feeling: Measurement of Sonic User Experience and Psychophysiology in a First-person Shooter Game. *Interact. Comput.* (New York, NY, USA) 22, no. 5 (Sept.): 336–343. ISSN: 0953-5438, doi:10.1016/j.intcom.2010.04.005, http://dx.doi.org/10.1016/j.intcom.2010.04.005.

Nacke, Lennart E., and Craig A. Lindley. 2008. Flow and immersion in first-person shooters: measuring the player's gameplay experience. In *Proceedings of the 2008 conference on future play: research, play, share,* 81–88. Future Play '08. New York, NY, USA: ACM. ISBN: 978-1-60558-218-4, doi:10.1145/1496984.1496998, http://doi.acm.org/10.1145/1496984.1496998.

Nacke, Lennart E., Craig A. Lindley, and Sophie Stellmach. 2008. Log Who's Playing: Psychophysiological Game Analysis Made Easy through Event Logging. In *Proceedings of the 2nd international conference on fun and games,* 150–157. Berlin, Heidelberg: Springer-Verlag. ISBN: 978-3-540-88321-0, doi:10.1007/978-3-540-88322-7_15, http://dx.doi.org/10.1007/978-3-540-88322-7_15.

Navara, Erika D., Edward O'Connor, Steve Faulkner, Robin Berjon, Travis Leithead, and Silvia Pfeiffer. 2014. *HTML5.* Candidate Recommendation. W3C.

Pagulayan, Randy J., Kevin Keeker, Dennis Wixon, Ramon L. Romero, and Thomas Fuller. 2003. User-centered Design in Games. In *The human-computer interaction handbook,* ed. Julie A Jacko and Andrew Sears, 883–906. Hillsdale, NJ, USA: L. Erlbaum Associates Inc. ISBN: 0-8058-3838-4, http://dl.acm.org/citation.cfm?id=772072.772128.

Parmar, Gunjeet, and Michael Thornburgh. 2012. *RTMP Specification 1.0.* Technical report. Adobe Systems Inc. http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf.

Russell, James A. 1980. A circumplex model of affect. *Journal of personality and social psychology* 39 (6): 1161.

Schulzrinne, Henning, Stephen L. Casner, Ron Frederick, and Van Jacobson. 1996. *RTP: A Transport Protocol for Real-Time Applications.* http://www.ietf.org/rfc/rfc1889.txt.

Seif El-Nasr, Magy, Heather Desurvire, Lennart Nacke, Anders Drachen, Licia Calvi, Katherine Isbister, and Regina Bernhaupt. 2012. Game User Research. In *Chi '12 extended abstracts on human factors in computing systems,* 2679–2682. CHI EA '12. New York, NY, USA: ACM. ISBN: 978-1-4503-1016-1, doi:10.1145/2212776.2212694, http://doi.acm.org/10.1145/2212776.2212694.

Shao, Bin, Du Li, Tun Lu, and Ning Gu. 2011. An Operational Transformation Based Synchronization Protocol for Web 2.0 Applications. In *Proceedings of the acm 2011 conference on computer supported cooperative work,* 563–572. CSCW '11. New York, NY, USA: ACM. ISBN: 978-1-4503-0556-3, doi:10.1145/1958824.1958910, http://doi.acm.org/10.1145/1958824.1958910.

Silbernagl, Stefan, and Agamemnon Despopoulos. 2007. *Taschenatlas Physiologie.* Thieme Electronic Book Library. Thieme. ISBN: 9783135677071.

Sing, Li. 2013. *Instant web applications with Meteor.* http://www.ibm.com/developerworks/opensource/library/wa-meteor-webapps/wa-meteor-webapps-pdf.pdf (accessed Jan. 18, 2014).

Wallner, Günter. 2013. Play-Graph: A Methodology and Visualization Approach for the Analysis of Gameplay Data. In *8th international conference on the foundations of digital games (fdg 2013),* 253–260.

Wallner, Günter, and Simone Kriglstein. 2012. A spatiotemporal visualization approach for the analysis of gameplay data. In *Proceedings of the 2012 acm annual conference on human factors in computing systems,* 1115–1124. ACM.

———. 2014. PLATO: A visual analytics system for gameplay data. *Computers & Graphics* 38:341–356.

Wehbe, Rina R., Dennis L. Kappen, David. Rojas, Matthias. Klauser, Bill. Kapralos, and Lennart E. Nacke. 2013. EEG-based Assessment of Video and In-game Learning. In *Chi '13 extended abstracts on human factors in computing systems,* 667–672. CHI EA '13. New York, NY, USA: ACM. ISBN: 978-1-4503-1952-2, doi:`10.1145/2468356.2468474`, `http://doi.acm.org/10.1145/2468356.2468474`.

Zammitto, Veronica. 2011. *The Science of Play Testing: EA's Methods for User Research.* Presentation at Game Developers Conference (GDC) 2011. `http://www.gdcvault.com/play/1014552/The-Science-of-Play-Testing`.

Zammitto, Veronica, Pejman Mirza-Babaei, Ian Livingston, Marina Kobayashi, and Lennart E. Nacke. 2014. Player Experience: Mixed Methods and Reporting Results. In *Chi '14 extended abstracts on human factors in computing systems,* 147–150. CHI EA '14. New York, NY, USA: ACM. ISBN: 978-1-4503-2474-8, doi:`10.1145/2559206.2559239`, `http://doi.acm.org/10.1145/2559206.2559239`.

Glossary

acetylcholine is a neurotransmitter of the parasympathetic system; apart from that, it also mediates the activation of eccrine sweat glands by sympathetic nerval endings. 50

autonomous nervous system (also known as vegetative, visceral or involuntary nervous system) is the collective term for an organism's sympathetic and parasympathetic nervous systems. 50

corrugator supercilii a facial muscle, causing a frowning face, when activated. 50

document object model is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML and XML documents. 50

dynamic data protocol is a data protocol, specialized on real-time data synchronization between servers and clients; it is developed by the Meteor Development Group for use in Meteor.js, an open source web application development framework. 50

eccrine sweat glands is the major type of sweat glands, found in the human skin, producing a clear, salty and odorless fluid. 50

electrodermal activity is measure for an organism's electric conductance on the surface of its skin; it can be measured using endosomatic (without an external electric current) and exosomatic methods (with an external electric current, see GSR). 50

electroencephalography a psychophysiological technique to measure the electric potential of different scalp regions over time to infer the underlying brain activity of different neocortical areas. 50

electromyography a physiological technique to measure the electric potential of one or more muscles over time. 50

first-person shooter is a video game genre, in which the player navigates through a three-dimensional environment, shooting at opponents from a first-person perspective. 50

galvanic skin response (also known as skin conductance (SC), electrodermal response (EDR), psychogalvanic reflex (PGR), skin conductance response (SCR) or skin conductance level (SCL)) is a technique to measure an organism's skin conductance applying external electric currents; it is directly related to the sympathetic nervous system and the activity of the skin's sweat glands. 50

games user research is a discipline of human-computer interaction, dealing with the evaluation of playability and gameplay experience in video games. 50

geographic information system is an information system, used to capture, store, edit, organize, analyze and present geographic data. 50

heart rate is the frequency of an organism's heartbeat, measured in numbers per unit of time, typically in beats per minute (bpm). 50

hemispheric frontal alpha asymmetry a psychophysiological metric, calculated from an electroencephalic recording (see EEG), in which the difference in Alpha band (8 - 15 Hz) power is compared between the two sides of the brain. 50

Hertz is the unit of occurences during a 1-second interval in the International System of Units. 50

human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (definition from SIGCHI Curricula, 1992). 50

HyperText Markup Language is a language, used to define the layout structure of a web page. 50

HyperText Transfer Protocol is a network protocol, used for the exchange of hypertext documents (like HTML documents) between two nodes in a computer network. 50

JavaScript is a scripting language, which is mainly, but not exlusively used in web browsers to dynamically manipulate web pages. 24, 27–29, 33, 36, 50

Javascript object notation is a syntax for the definition of hierarchical data objects, used primarily, but not exclusively in the scripting language Javascript. 50

level-of-detail is a process of simplifying the representation of an object by leaving out details (i.e., parts of it, which are considered less important than others) in order to improve the speed of representation. 50

neurotransmitter is a biochemical molecule with the function to transmit a neural signal from one nerve cell to another cell. 9, 50

ohm (symbol Ω) is the unit of electric resistance in the International System of Units. 50

operational transformation is the name for a set of algorithms, facilitating collaborative editing of content by multiple users by offering advanced techniques of write conflict resolution. 50

psychophysiology is a scientific discipline that deals with the physiological bases of psychological phenomena such as emotional and cognitive processes. 6, 50

siemens (also mho) is the unit of electric conductance in the International System of Units. 50

skin conductance level is a synonym of electrodermal acitvity (see EDA). 50

sympathetic nervous system is a part of an organism's autonomic nervous system; its function is to activate the organism's general level of stress and to activate its fight-or-flight response. 50

Transfer Control Protocol is a connection-oriented network protocol, used for the reliable exchange of data streams between two nodes in a computer network. 50

user-centered design is an iterative design process for interactive products and systems, in which the user, his tasks and environment play a central role. 50

zygomaticus major a facial muscle, involved in the generation of smiles. 50

Appendices

A Averaging Aggregations in MongoDB

```
var aggregation = db.timeSeriesPoints.aggregate([
    { $match: {
        "value.s": options.timeSeriesId,
        "value.l": options.inputLevel,
        "value.t": { $gte: options.startTimePosition }
    }},
    { $sort: {
        "value.t": 1
    }},
    { $limit: options.numberOfPointsToAggregate
    },
    { $project: {
        s: "$value.s",
        t: { $subtract: [
            "$value.t",
            { $mod: [ "$value.t", options.outputLevel ] }
        ] },
        ts: { $substr: [ { $subtract: [
            "$value.t",
            { $mod: [ "$value.t", options.outputLevel ] }
        ] }, 0, 128 ] },
        a: { $multiply: [ "$value.a", "$value.c" ] },
        l: { $add: [ options.outputLevel ] },
        ls: { $substr: [ options.outputLevel, 0, 128 ] },
        c: "$value.c"
    }},
    { $group: {
        _id: "$t",
        s: { $first: "$s" },
        ts: { $first: "$ts" },
        a: { $sum: "$a" },
        l: { $first: "$l" },
        ls: { $first: "$ls" },
        c: { $sum: "$c" }
    }},
    { $sort: {
        _id: 1
    }},
    { $project: {
        _id: { $concat: [ "s_", "$s", "_l_", "$ls", "_t_", "$ts" ] },
        "value.s": "$s",
        "value.t": "$_id",
        "value.a": { $divide: [ "$a", "$c" ] },
        "value.l": "$l",
        "value.c": "$c"
    }}
]);
if (aggregation.ok == 1 && aggregation.result.length > 0) {
    db.timeSeriesPoints.insert(aggregation.result);
}
```

Listing 1: The MongoDB commands to perform an averaging aggregation on time series data, using MongoDB's built-in aggregation pipeline.

```
db.runCommand({
    mapReduce: "timeSeriesPoints",
    query: {
        "value.s": options.timeSeriesId,
        "value.l": options.inputLevel
    },
    map: function() {
        var point = this.value;
        point.l = options.outputLevel;
        var startTime = point.t - point.t % options.outputLevel;
        var id = "s_" + options.timeSeriesId +
                 "_l_" + options.outputLevel +
                 "_t_" + startTime;
        emit(id, point);
    },
    reduce: function(id, points) {
        var summedAmplitude = 0;
        var totalCount = 0;

        var numberOfPoints = points.length;
        for (var i = 0; i < numberOfPoints; ++i) {
            var point = points[i];
            summedAmplitude += point.a * point.c;
            totalCount += point.c;
        }

        var averagedAmplitude = summedAmplitude / totalCount;
        var startTime = points[0].t;

        var aggregatedPoint = {
            s: options.timeSeriesId,
            t: startTime,
            a: averagedAmplitude,
            l: options.outputLevel,
            c: totalCount
        };

        return aggregatedPoint;
    },
    out: {
        merge: "timeSeriesPoints",
        nonAtomic: true
    }
});
```

Listing 2: The MongoDB command to perform an averaging aggregation on time series data, using MongoDB's built-in MapReduce framework.