

Uwe Kloos, Gabriela Tullius (Hrsg.)

Informatics Inside:

Mensch.Maschine.Immersion.

Virtualität trifft Realität

Informatik Konferenz an der Hochschule Reutlingen

05. Mai 2010



INF

Studiengang
Medien- und
Kommunikationsinformatik



Hochschule Reutlingen
Reutlingen University

Impressum

Anschrift

E-Mail: infoinside@reutlingen-university.de
Internet: <http://www.infoinside.reutlingen-university.de/>

Hochschule Reutlingen
Fakultät Informatik
Medien- und Kommunikationsinformatik
Alteburgstraße 150
D-72762 Reutlingen

<http://www.mki.reutlingen-university.de/>
Telefon: +49 7121 271 – 4002
Telefax: +49 7121 271 – 4042

Organisationskomitee

Prof. Dr. Uwe Kloos, Hochschule Reutlingen
Prof. Dr. Gabriela Tullius, Hochschule Reutlingen
Danja Walz, Hochschule Reutlingen

Achim Lang	Manuel Mattes
Armin Wälder	Marc Ammon
Christian Neth	Michael Bierkandt
Christian Veizhans	Niky Metzger
Diana Hauser	Patricia Duensing
Emre Yay	Philipp Becker
Ivelina Alexandrova	Thomas Lauria
Janina Bierkandt	Ulf Maier

Copyright: Creative Commons Namensnennung 3.0 Deutschland Lizenzvertrag
Dieses Werk ist unter einem Creative Commons Namensnennung 3.0 Deutschland Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/3.0/de/> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Druck: Friedmann* Print Data Solutions

Vorwort

Nach dem erfolgreichen Start im letzten Jahr wird die Informatik Konferenz 'Informatics Inside' zum zweiten Mal an der Hochschule Reutlingen durchgeführt; 2010 lautet das Schwerpunktthema "Mensch.Maschine.Immersion. Virtualität trifft Realität".

Informatics Inside wurde im Studiengang Medien- und Kommunikationsinformatik kreiert: Masterstudenten organisieren eine Konferenz, an der sie Themen aus ihrer Ausbildung präsentieren und zur Diskussion stellen. Der wissenschaftliche Diskurs mit aktuellen Themen aus der Informatik soll Studenten und Fachleuten die Gelegenheit geben, sich um Inhalte zu streiten und Neues zu erfahren. Mit diesem Forum wollen die Studenten ihre Hochschulerfahrungen dem harten Praxistest unterziehen. Diese Konferenz richtet sich nicht nur an Studenten - herzlich eingeladen sind alle Fachleute und Informatiker aus Wirtschaft, Wissenschaft und Forschung, Vertreter der Industrie und natürlich Studenten und Professoren aller Hochschulen und alle interessierten und neugierigen Gäste.

Als Gastredner begrüßen wir Dr. Betty J. Mohler, vom Max Planck Institut für biologische Kybernetik in Tübingen und Dr.-Ing. Christoph Runde, vom Virtual Dimension Center (VDC) in Fellbach.

Begleitend zur Konferenz werden studentische Arbeiten und Projekte präsentiert und an den Informationsständen der Industrie können Besucher einen Blick hinter die Kulissen der IT-Branche werfen. Ein gutes halbes Jahr an Vorplanung, Gesprächen und Testforen liegen hinter uns. Dank an dieser Stelle allen engagierten Studenten des Seminars "Wissenschaftliche Vertiefung" und allen anderen Unterstützern.

In diesem Sinne wünsche ich Ihnen viel Spaß beim Informationsaustausch und bei kritischen Diskussionen.



Prof. Boris Terpinc
Studiengangsleiter Medien- und Kommunikationsinformatik

Inhaltsverzeichnis

Egozentrische Entfernungsschätzungen in virtuellen immersiven Umgebungen mit unterschiedlicher Beleuchtung	7
Ivelina Alexandrova	
Der Einfluss der Gehgeschwindigkeit auf Redirected Walking.....	15
Christian Neth	
Echtzeitcomputergrafik - Interaktive virtuelle Welten im Entertainmentbereich	24
Achim Lang	
Service-orientierte Architekturen für die Integration von Ingenieurprogrammen im Rahmen der digitalen Produktentwicklung.....	30
Thomas Lauria	
Web 3D - Anwendungsgebiete und Technologien	38
Armin Wälder, Ulf Maier	
Adaptive Systeme - Gestaltung und Einsatz.....	55
Janina Bierkandt, Michael Bierkandt	
Konzeption eines dezentralen sozialen Netzwerkes basierend auf offenen Webtechnologien.....	74
David Schlichtenberger	
Modell eines TOGAF basierten Portals für Enterprise Architecture Management (EAM)	78
Matthias Gutbrod, Felix Schiele, Christian Zinsenhofer	
Hoardingverfahren für eine Context-Aware Kalender Applikation	83
Roman Preuß, Rouven Alexander Rieker, Aleksandr Goncharov, Kateryna Shavrytska, Nazmi Tachiroglu, Dirk Marciniak	
Partner der Konferenz.....	89

Egozentrische Entfernungsschätzungen in virtuellen immersiven Umgebungen mit unterschiedlicher Beleuchtung

Ivelina Alexandrova

Max-Planck-Institut für biologische Kybernetik
Medien- und Kommunikationsinformatik, Hochschule Reutlingen
Ivelina.Alexandrova@student.Reutlingen-University.DE

Betreuer: Prof. Dr. Uwe Kloos, Hochschule Reutlingen
Dr. Betty J. Mohler, Max-Planck-Institut für biologische Kybernetik
Prof. Dr. Heinrich H. Bülthoff, Max-Planck-Institut für biologische Kybernetik

Abstract:

Im Vergleich zu den Entfernungsschätzungen in der Realität, werden die egozentrische Entfernungsschätzungen in virtuellen immersiven Umgebungen, die in Head-Mounted Displays projiziert sind, unterschätzt. Viele Wissenschaftler haben versucht, der Grund dafür herauszufinden. Obwohl er noch nicht bekannt ist, zeigt eine Studie, dass die mündlichen Entfernungsschätzungen von der Qualität der Computergraphik beeinflusst sind. Der Begriff „Qualität der Computergraphik“ umfasst einige Elemente, wie z. B. Beleuchtung und räumliche Relation. In diesem Zusammenhang, beschreibt diese Arbeit der Anfang einer Studie über die Auswirkungen der unterschiedlichen Beleuchtungen in virtuellen Welten auf den egozentrischen Entfernungsschätzungen. Dazu wird es gezeigt, wie ein 3D Modell für die Ziele der Studie entwickelt ist, der auch für andere Forschungen benutzt werden kann. Außerdem wurde der Pilotversuch der Studie präsentiert.

1 Motivation and Related Work

Nowadays, immersive virtual environments (VEs) are used for training and educational purposes. They are also a promising tool for the architecture and the industry. Showing a costumer a 3D model of a building or a car, which are not yet built or produced, gives a better impression of the design of the future product. Therefore, at present the aim of the virtual worlds' creators is to build them as realistic as possible.

However, a lot of studies show that virtual worlds appear to the user smaller than they really are. This compression could be up to 50% (Philbeck & Loomis 1997;

Loomis & Knapp 2003). Several response measures, as for example, verbal reports, blind walking, pointing and throwing, have been used in order to determine a person's egocentric distance perception. There are a great number of studies examining this problem and in particular, what is the reason that causes this underestimation of distances. According to some studies these inaccuracies could be due to the fact that usually people do not see themselves in the virtual world and do not have a spatial relation. In addition, the response measure, the feeling of presence, the field of view (FOV), the weight/inertia of the head-mounted display (HMD), and the quality of the computer graphics, have all been investigated as possible causes.

Of most interest to the present paper, initially Thompson et al. suggested that the quality or realism of graphics does not have an impact on distance judgments in VEs (Thompson et al. 2004). More recently, Kunz et al. demonstrate that quality of graphics does not influence a response involving locomotion, but it does influence verbal reports of egocentric distances (Kunz et al. 2009). Quality of computer graphics could refer to several different elements of graphics, such as increased number of objects or realism of lighting.

In our study we want to examine, whether verbal reports are influenced by the quality of the lighting in a scene. For the purpose of this study a realistic 3D model of a real world room was modeled and different versions of it were used for conducting a pilot experiment. Each version uses a different lighting technique. We suppose that the better lighting conditions may convey a feeling of presence in the virtual world and may provide the user with the details necessary to divide the space. In order to examine the effect of the different lighting techniques we conducted a pilot experiment in both a HMD VE and a panoramic display VE.

The work in this paper shows the first step towards an extensive research, which final goal is to examine, whether lighting does have an impact on people's distance judgments. Moreover, we want to study, how the different lighting of the VE combined with different displaying techniques will affect distance judgments. In the following section the 3D models and the setup used for the study, are introduced. The pilot experiment and some preliminary results are also presented.

2 Stimuli and Apparatus

In this part the 3D model of the real room and its different versions are presented in detail. In addition, the setup used for the experiment is introduced.

2.1 3D Models

For the purpose of the study a 3D model of a real room, located near to the tracking hall and the panoramic display, was modeled. Therefore, the experiments can be conducted in both the virtual and the real world and the participant's performance can be compared. The 3D model together with its content has 10,488 polygons and is modeled in Autodesk 3ds Max 2009. Thus, it can be used not only for this experiment, but also as a surrounding environment for other real-time experiments, such as Dodds et al.'s experiment (see Figure 1).



Figure 1: Left: Participants tracked in the tracking hall. Right: Participants' avatars in the 3D model in real-time. (Dodds et al. 2010)

To give familiar size cues, the author has modeled tables, chairs, etc. The dimensions of the 3D room are the same as in the real one (10.27m x 7.25m x 2.77m). To make the materials in the scene as realistic as possible, the textures were extracted from photos of the real room. In addition, the textures were done in a way that the user can not notice the repeating patterns of the carpet or the ceiling, for instance. To examine, whether lighting has an impact on people's egocentric distance judgments, the models vary only in the presence of light cues. Therefore, familiar size cues and nice textures are present in each of the three models used for the research (Figure 2):

- *Model with global lighting* (Figure 2 bottom right). – In this model only the global lighting of Autodesk 3ds Max 2009 is used. Therefore, no additional light source is set up. For this reason, there are no objects that cast or receive shadow. Consequently, with respect to light cues, this model is less realistic than the other two used for the experiment. We suppose that in this environment participants will compress the space more than in the other two models.
- *Model with ambient occlusion lighting* (Figure 2 bottom left). – This model

uses the *model with global lighting* as a basis. Thus, it has the same familiar size cues and textures. The only difference between this model and the previous one is the different light cue. For creating better lighting in this model a global illumination technique, called ambient occlusion, was used. It generates the type of deep shadow that appears in the corners of objects.

Ambient light does not come from a particular light source, and therefore no additional light source is needed. For better real-time performance the ambient occlusion light of the scene is rendered on ambient occlusion maps. They were rendered only for the objects that have an impact on the scene, such as the carpet or the walls. We suppose that due to the shadows, users should be able to judge distances better in this model than in the one with global lighting.

- *Model with simulated sun light* (Figure 2 top right) – In this model the *model with global lighting* is again used as a basis. Additionally, for creating realistic lighting, a sun light system source, was adjusted and set up to simulate the sun. Then, for the sake of better real-time performance, the light coming from the sun light system was rendered on light maps, where the familiar size cues were also taken into account.



Figure 2: Top left: Real world room. Bottom left: Model with ambient occlusion lighting. Top right: Model with simulated sun light. Bottom right: Model with global lighting.

Thus, shadows are cast as they would appear in the real world. In terms of lighting this model is the most realistic one, which should help the participants

to feel present in the space. Therefore, we expect them to perform better in this model than in the previous two.

2.2 Setup

After the different versions of the 3D model were created in Autodesk 3ds Max 2009, they were exported in Virtools 4.1, a Dassault Systemes' development platform for creating interactive 3D real-time applications. The materials used for the 3D models are Shell materials and were exported as such in Virtools 4.1. This enables manipulation of the materials in Virtools 4.1 itself. Furthermore, a particular object can have a tiled texture, while a light map is wrapping the whole object, as for example the floor, the ceiling or the walls. Moreover, in Virtools 4.1 the texture of the chairs, for instance, can be assigned to the carpet, while the carpet keeps its own ambient occlusion map. In addition, the program which changes the target location and blanks the screen between trials is also written in Virtools 4.1.



Figure 3: Top left: participant in the HMD VE. Bottom left: the 3D model projected in the HMD. Right: Participant in the panoramic display VE.

For the experiment conducted in HMD VE (Figure 3 top/bottom left) we have used an nVisor SX60 HMD with a resolution 1280x1024 for each eye and FOV of 44 degrees. The experiment itself took place in a tracking hall, which is 12x12m in size. Users' head position and orientation are tracked using monitoring of reflective markers and an optical tracking system (16 Vicon MX13 cameras that have LED strobe fixed lights). The Vicon system records the movements in the real-world and reproduces them in the VE. Each Vicon camera has a resolution of 1280x1024. The tracking system has a maximum frame rate of 484Hz. The measured latency of the system is approximately 30ms.

In our experiment we do not track the whole body of the participant, because we are not interested in his or her movements. Moreover, in our further research we

will be rather interested in the walked distance. Therefore, we track only the five markers of the HMD. They are ordered in a way that they do not occlude each other, which makes them always visible for the Vicon system.

The second part of the experiment took place in a panoramic display VE (Figure 3 right). The system uses four JVC D-ILA DLASX21S video projectors each with resolution of 1400x1050 pixels. These projectors are used to display visual stimuli on the front, the sides, and the floor of a custom made curved panoramic display. It has FOV of 220 degrees horizontal by 165 degrees vertical. For the experiment participants stand at a distance of 3.5m from the curved display.

3 Procedure and Experimental Design

For conducting the pilot experiment we have used only two versions of the 3D model. The experiment had a between-subjects design and took place in both the HMD VE and the panoramic display VE. Participants were asked to make egocentric verbal judgments of distances in either one of the following two parts of the pilot experiment:

- HMD VE, where participants saw first *the model with global lighting* and then *the model with simulated sun light*. Participants' hands and feet were not tracked and they were not able to see themselves in the virtual room.
- Panoramic display VE, where participants saw first the *model with simulated sun light* and then the *model with global lighting*. Participants' were able to see themselves, because the virtual room is projected on a large screen in front of them.

Each part of the pilot experiment consisted of 19 trials. The participants did first 3 pilot trials, where they learn how to make verbal judgments of distances, and then they did eight trials per model. For each trial a green target was placed on the floor of the virtual room in front of the participant. The possible distances, in which the target was placed, were eight – 2m, 2.5m, 3m, 3.5m, 4m, 4.5m, 5m and 5.5m. The order, in which they appear, was randomly generated, but each one appeared ones per model. The task, that the participants had, was to determine as accurate as possible how far from them the center of the target is. The response measure used in both VEs was verbal reports.

Before the beginning of the experiment the participants were given written and oral instructions how to make a verbal judgment of egocentric distances (Kunz et al. 2009). During the experiment when participants were seeing the virtual room they were not allowed to walk or lean. Although, they were only permitted to

move their head about their neck, they were encouraged to take as much time as they need to look around and get familiar with the environment. They had to give a sign to the experimenter, when they think they can imagine with closed eyes, where exactly the object is located. Then the experimenter blanked the screen of the HMD or the panoramic display. After that participants had to turn to the left and call out how far from them the center of the target is located. In order to start the next trial they had to turn back to their initial position. After that, they were able to see the virtual room again. The experiment took about 30 min per participant.

4 Preliminary Results

This section provides some preliminary results of this pilot experiment. The graph in Figure 4 (left) shows the results of a participant A that took part in the HMD VE. Compared to the real distance, there is an underestimation in both models.

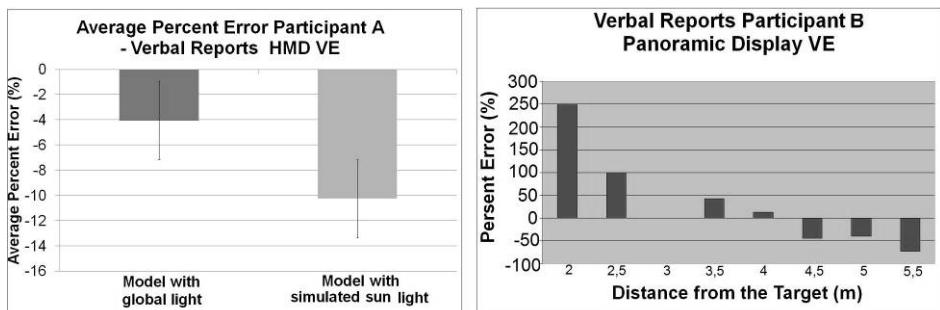


Figure 4: Left: Participant A's average error per model in the HMD VE. Error bars represent one standard error from the mean. **Right:** Participant B's verbal reports panoramic display VE.

Interesting is that the participant gave better results on the model with global lighting (4% underestimation), while in the model with better lighting the participant underestimated distances by 10%. Of course, deeper analysis has to be done to find out not only whether light cues help the user to divide the space better, but also what is the amount of lighting needed for accurate verbal judgments of distance in VEs.

In Figure 4 (right), the results of participant B in the panoramic display present an overestimation of distances that are less than 4m, while for distances beyond 4m there is an underestimation. Interesting is the fact that the curve of the panoramic display was at distance of 3.5m from the participants. Perhaps there could be a

conflict between the dimensions of the panoramic display and the 3D model used for conducting the experiment. However, further investigation is needed, in order to make any final assumptions for this experiment.

5 Conclusion

This paper presented the motivation, the preparation and the beginning of a research, which final goal is to examine, whether light cues provide the user with the details necessary to divide the space into more accurate units, and therefore have an impact on people's distance judgments. Moreover, we are going to investigate, whether familiar size cues have stronger impact on egocentric judgments of distances than light cues. Furthermore, we will do a deeper analysis for the panoramic display VE, to determine the cause of the underestimation.

Bibliography

- Dodds T.; Mohler B. J.; Bülthoff H. H. (2010): A Communication Task in HMD Virtual Environments: Speaker and Listener Movement Improves Communication. To appear in CASA.
- Kunz B. R.; Wouters L.; Smith D.; Thompson W. B.; Creem-Regehr S. H. (2009): Revisiting the effect of quality of graphics on distance judgments in virtual environments: A comparison of verbal reports and blind walking. *Attention, Perception, & Psychophysics* 71, 6, 1284–1293.
- Loomis J. M.; Knapp J. (2003): Visual perception of egocentric distance in real and virtual environments. In *Virtual and Adaptive Environments*, Hettinger L. J., Haas M. W., (Eds.). Erlbaum, Mahwah, NJ. Chap. 2, 21–46.
- Philbeck J. W.; Loomis J. M. (1997): Comparison of two indicators of perceived egocentric distance under full-cue and reduced-cue conditions. *Journal of Experimental Psychology: Human Perception and Performance* 23, 1, 72–85.
- Thompson B.; Willemse P.; Gooch A. A.; Creem-Regehr S. H.; Loomis J. M.; Beall a. C. (2004): Does the quality of the computer graphics matter when judging distances in visually immersive environments? *Presence: Teleoperators and Virtual Environments* 13, 5, 560–571.

Der Einfluss der Gehgeschwindigkeit auf Redirected Walking

Christian Neth

Medien- und Kommunikationsinformatik, Hochschule Reutlingen
Christian.Neth@student.hochschule-reutlingen.de

Betreuer: Dr. Jan Souman, MPI Tübingen
Dr. Betty Mohler, MPI Tübingen
Prof. Dr. Heinrich Bülthoff, MPI Tübingen
Prof. Dr. Uwe Kloos, Hochschule Reutlingen

Abstract:

Wenn virtuelle Welten durch Gehen erkundet werden stellt die räumliche Beschränkung der realen Gehfläche oftmals ein Hindernis dar. Umgeleitetes Gehen (*Redirected Walking*) stellt eine Technik dar, mit deren Hilfe der virtuell begehbar Bereich über den real vorhandenen Raum ausgedehnt werden kann. Neben anderen Möglichkeiten kann hierbei ein Kurvenbiegungsfaktor (*curvature gain*) angewandt werden, der virtuelle Geraden auf reale Kreisbögen abbildet. Durch geeignetes Einsetzen der Kurvenbiegung wird versucht, den Benutzer stets von den physikalischen Grenzen fern zu halten.

Da verwandte Arbeiten einen Einfluss zur Gehgeschwindigkeit vermuten, untersucht diese Arbeit den Effekt der Gehgeschwindigkeit auf Kurvensensibilität. Testpersonen gingen in einer virtuellen Welt verschieden schnell auf bestimmten Kreisbögen und gaben Rückmeldung über die Richtung der gegangenen Kurve. Die Ergebnisse zeigen, dass die Unterscheidungsgrenze für langsamere Gehgeschwindigkeiten sinkt.

1 Einleitung und verwandte Arbeit

Das Gehen stellt für Menschen die intuitivste und natürlichste Art dar, sich durch Räume zu bewegen und die Umgebung zu erkunden. Dies ist auch für virtuelle Welten, die mit Hilfe einer Datenbrille (*head-mounted display, HMD*) betrachtet werden, der Fall. Hierbei ergeben sich durch die räumlichen Grenzen der in der realen Welt zur Verfügung stehenden Gehfläche eine Beschränkung der potentiell „unendlich“ großen virtuellen Welt.

Um diese Beschränkung zu verringern oder aufzuheben beschreiben Razzaque et al. (2001) eine Technik, die die Abbildung der realen Position des Benutzers zur virtuellen Position verändert. Diese Technik, das sogenannte 'Umgeleitete Gehen'

(*Redirected Walking*), macht sich zunutze, dass Menschen beim Gehen relativ unempfindlich gegenüber Rotation und Kurven sind (Cratty & Williams 1966; Kallie et al. 2007) sowie die Entferungen in virtuellen Welten schlecht einschätzen können (Nitzsche et al. 2004). Somit ist es innerhalb bestimmter Grenzen möglich, während Rotationen oder Translationen den Benutzer in der realen Welt so umzuleiten, dass der vorhandene Raum besser genutzt wird (vgl. Abb. 1). Steinicke et al. (2009) beschreiben fünf Komponenten, anhand derer eine Manipulation der Positionsabbildung ebenfalls möglich ist:

- Skalierung der Translation (*Translation gain*)
- Skalierung der Rotation (*Rotation gain*)
- Zuführen von Kurvenbiegung (*Curvature gain*)
- Zuführen von lateralem Versatz während Rotationen (*Displacement gain*)
- Zeitabhängige Zuführung von Rotation (*Time-dependent gain*)

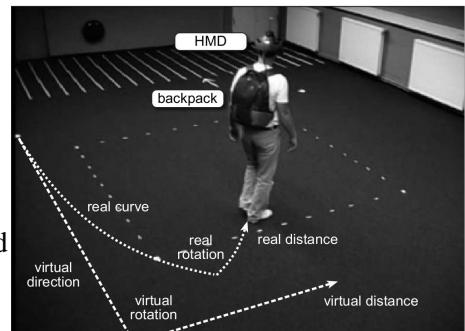


Abbildung 1: Umgeleitetes Gehen (Steinicke et al. 2009)

Engel et al. (2008) führten den Benutzer auf einem geschlängelten Pfad durch eine virtuelle Stadt und zeigte dadurch, dass durch den Einsatz der Rotationsskalierung ein Gehen in sehr großen virtuellen Umgebungen mit vorgegebenen Rotationen möglich ist. Durch Skalierung der Rotation in Abhängigkeit zum Abstand des Benutzers zur Wand konnte die -im Ganzen gesehen- gerade Bewegung in einen gezackten Kreis verändert werden. Verschiedene andere Arbeiten beschreiben eine quantitative Messung der möglichen Radien für eine Fehlleitung auf eine Kurve. Kallie et al. (2007) beschreiben, dass Kurven bis einem Radius von ca. 25m nicht zuverlässig erkannt werden konnten, während Cratty & Williams (1966) sowie Steinicke et al. (2009) deutlich kleinere Werte von ca. 5,5m bzw. 6m angeben. Steinicke et al. (2009) deutet an, dass die Größe der Kurvenskalierung von der Gehgeschwindigkeit abhängig sein könnte. Unsere Studien untersuchten daher, ob die Gehgeschwindigkeit des Benutzers einen Einfluss auf die mögliche zuführbare Kurvenbiegung hat. Hierzu modellierten wir eine virtuelle Welt, in der Testpersonen einer Kugel folgten. Die Kugel bewegte sich in verschiedenen bestimmten Geschwindigkeiten auf Kreisbögen verschiedener Radien. Die Testpersonen mussten bestimmen, ob sie auf einer Links- oder Rechtskurve gegangen waren.

2 Versuchsaufbau

2.1 Labor, Hardware & Software

Das Labor, in dem der Versuch stattfand, hat eine Fläche von ungefähr 12x12m. Wegen Begrenzungen des Trackingsystems und um einen Sicherheitsabstand zur Wand zu wahren, fand der Versuch auf einer Fläche von ca. 10x10m statt.

Die Position und Kopfausrichtung der Testperson wurde durch ein optisches Trackingsystem (16 ViconMX13 Kameras) erfasst. Jede Kamera hat eine Auflösung von 1280x1024 Pixeln bei einer maximalen Bildwiederholrate von 484 Hz. Die gemessene Latenz des Systems liegt bei ca. 30ms. Die Position und Kopfausrichtung der Testperson wurde mit der selben Frequenz aufgezeichnet und für die spätere Analyse mittels MATLAB in Logdateien gespeichert.

Die virtuelle Umgebung wurde mit Virtools 4.1 erstellt und auf einem Dell Inspiron M1710 Laptop (2.16 GHz T7400 CPU; 2 GB RAM; NVIDIA GeForce Go 7950 GTX Grafikkarte) mit einer Auflösung von 800x600 Pixeln bei einer Bildwiederholrate von 60 Hz ausgeführt. Als Ausgabegerät verwendeten wir ein eMagin 3D Visor (FOV 32°x24°).

Um ungewöhnliche Massenträgheit zu vermeiden und um das Experiment für die Testperson angenehmer zu gestalten, wurde alle Ausrüstung außer HMD, Trackinghelm, Kopfhörern und einem Spielecontroller vom Experimentator getragen.

2.2 Gestaltung der virtuellen Welt

Um lediglich den Einfluss von Propriozeption, Vestibulärsystem und Efferenzkopien auf die Kurvenempfindlichkeit zu testen, wurde die virtuelle Umgebung sehr schlicht gehalten. Jedes existierende Element hatte einen bestimmten Zweck im Rahmen des Experiments.

2.2.1 Virtuelle Umgebung

Der Boden bestand aus einer quasi unendlichen Fläche, die mit einer Textur aus weißem Rauschen texturiert war. Um der Testperson keine Anhaltspunkte zur Rotation (z.B. durch Wolken oder die Sonne) zu geben, wählten wir statt einer 'richtigen' Himmelstextur lediglich ein gleichförmiges Blau. Um die Drehung der Testperson durch das Gehen auf einer Kurve auszugleichen, wurde der Boden entsprechend gedreht um so die Illusion einer Geraden zu erzeugen.

2.2.2 Enthaltene Elemente

In der virtuellen Umgebung wurde auf dekorative Elemente verzichtet. Die einzigen existierenden Elemente waren:

- Eine Kugel führte die Testperson in einer bestimmten Geschwindigkeit auf einer bestimmten Kreisbahn. Hierbei gab sie über Farbänderungen Rückmeldung über den Abstand zur Testperson; eine grüne Kugel signalisierte einen korrekten Abstand. Wurde der Abstand zu groß oder zu klein, ändert die Kugel ihre Farbe über Gelb nach Rot.
- Ein blauer Zylinder (1x2m) markierte die Startposition der Testperson.
- Ein zweiter Zylinder zeigte die gerade Gehrichtung an und gab -in gleicher Weise wie bei der Kugel- Rückmeldung über die korrekte Positionierung der Testperson innerhalb des blauen Zylinders. Um Verwirrungen zu vermeiden wurde der zweite Zylinder erst eingeblendet wenn der Benutzer im blauen Zylinder stand.
- Eine weiße Linie diente dazu, den Benutzer in eine bestimmte Richtung auszurichten. Dies war in zwei Punkten im Experiment wichtig:
 - Zu Beginn jeder Versuchsreihe wurde die Benutzerausrichtung kalibriert. Dazu wurde der Benutzer aufgefordert, an einen bestimmten Punkt im Raum zu gehen und sich entlang der weißen Linie auszurichten. So konnte ein schief sitzender Trackinghelm ausgeglichen werden.
 - Zu Beginn jedes Testlaufs diente die weiße Linie dazu, dem Benutzer die gerade Gehrichtung anzuzeigen. Sie begann dazu am blauen Zylinder und führte in Richtung des zweiten Zylinders (vgl. Abb. 2).

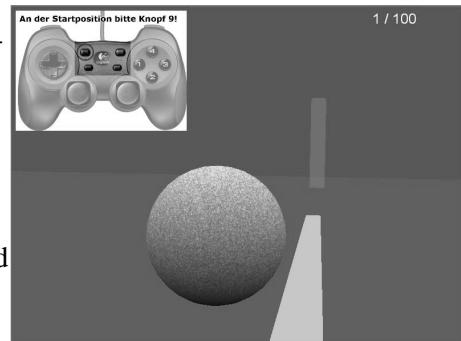


Abbildung 2: Ansicht beim Start

2.2.3 Startposition

Die Startpositionen, von denen die Testperson ihre Testläufe startet, waren prinzipiell für jeden Testlauf verschieden. Sie befanden sich auf einem Kreis mit 3,5m Radius um den Mittelpunkt des Raums. Obwohl der Raum quadratisch mit den Maßen von ca. 12x12m ist, wurde dieser Radius ausgewählt, da so die durchschnittliche Gehstrecke im optimalen Bereich des Trackingsystems liegt und dennoch für die längste Gehstrecke ausreichende Sicherheitsabstände zu den Wänden

vorhanden ist. Dieser Radius wurde durch das Experiment konstant gehalten. Um der Testperson keine Rückmeldung über die Korrektheit des letzten Testlaufs zu geben, wurde als nächste Startposition ein Punkt auf dem Kreis in gerader Richtung vor der Kugel gewählt. Die Gehrichtung zeigte stets von der Startposition zur Mitte des Raums.

Um eine „Rekalibrierung“ während des Aufsuchens der nächsten Startposition zu vermeiden, wurde während dieser Phase die Bodentextur durch eine einfarbige Textur ersetzt.

2.2.4 Anweisungen

Um die Menge von Anordnungen, die die Testperson im Gedächtnis behalten muss, zu reduzieren, wurden Anweisungen zum Versuchsablauf in der oberen linken Ecke der Anzeige angezeigt. Da die Testperson den Spielecontroller nicht sehen konnte, wurde ein Foto des Controllers dargestellt, auf dem die zu drücken- den Knöpfe graphisch hervorgehoben waren.

3 Versuchsdurchführung

3.1 Testpersonen

Am Versuch nahmen 10 Testpersonen (3 Frauen, 7 Männer) im Alter von 21 – 29 Jahren ($\bar{\Omega}$ 24,4) teil. Sie waren naiv zum Experiment und hatten gute oder korrigierte Sehschärfe. Für ihren Komfort wurde ihnen geraten, bequemes Schuhwerk zu tragen.

Um einen potentiellen Einfluss der Simulatorkrankheit (Kinetose) auf die Versuchsergebnisse aufzuzeichnen füllten alle Testpersonen vor und nach jeder Sitzung einen Fragebogen zur Simulatorkrankheit aus.

Um eine potentiell mögliche Orientierung an Umgebungsgeräuschen zu vermeiden, trug jede Versuchsperson geräuschdämpfende Kopfhörer, durch die weiße Rauschen gespielt wurde. Abb. 3 zeigt eine vollständig ausgerüstete Testperson.

Eine Versuchsperson zeigte nach dem Ver- such starke Symptome der Simulatorkrank- heit; ihre Werte flossen nicht in die endgültige Analyse ein.



Abbildung 3: Testperson

3.2 Versuchsbedingungen

Die Geschwindigkeiten der Versuchsreihen wurden - ausgeglichen gemischt - im Voraus festgelegt und in umgekehrter Reihenfolge wiederholt um den Trainingseffekt auszugleichen. Jede Testperson absolvierte zwei Versuchsreihen pro Geschwindigkeit. Die Geschwindigkeiten betrugen 0,75, 1,00 und 1,25 m/s.

Die Gehzeit wurde im Bereich zwischen 6 und 7 Sekunden zufällig gewählt, wodurch eine Konditionierung auf eine bestimmte Dauer vermieden werden soll.

In Pilotversuchen zeigte sich ein Bereich von sinnvollen Kurvenbiegungen zwischen 0,05 und 0,005 m⁻¹, was Radien von 20 bis 200m entspricht. Wir wählten 10 Kurvenbiegungen im Abstand von 0,005m⁻¹ und präsentierten jede in zufälliger Reihenfolge je 5x nach links und rechts. Somit ergaben sich 100 Testläufe pro Testperson pro Geschwindigkeit.

3.3 Durchführung

Der Versuch wurde pro Testperson in zwei Sitzungen durchgeführt. In jeder Sitzung wurden drei Versuchsreihen mit verschiedenen Geschwindigkeiten absolviert. In jeder Versuchsreihe musste vor den eigentlichen Testläufen ein Training absolviert werden, außerdem fand zu Beginn eine Kalibrierung des Trackinghelms statt.

Zu Beginn des Versuchs wurde in einem Fragebogen der Einfluss der virtuellen Präsenz auf die Testperson abgefragt sowie nach dem Versuch experimentspezifische Details wie gewählte Strategien.

3.3.1 Training

Das Training bestand aus zwei Phasen:

In der ersten Phase (vgl. Abb. 4) wurde das korrekte Verfolgen der Kugel geübt. Aufgabe der Testperson war es, den korrekten Abstand zur Kugel für 2 Minuten zu halten während sich die Kugel in einem Quadrat bewegte. Falls sich die Testperson noch nicht mit der Versuchsanordnung vertraut fühlte konnte diese Phase verlängert werden. Allerdings machte keine der Testpersonen von dieser Möglichkeit Gebrauch. In

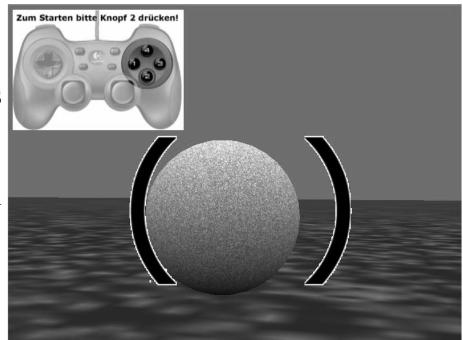


Abbildung 4: Erste Trainingsphase

der zweiten Trainingsphase mussten 10 'reale' Testläufe durchlaufen werden. Pilotversuche hatten ergeben, dass nach ungefähr 10 Testläufen der Abstand zur Kugel

deutlich besser eingehalten werden konnte. Die Daten dieser Testläufe flossen nicht in die Analyse mit ein.

3.3.2 Testläufe

Jeder Testlauf lief nach dem gleichen Muster ab:

1. Die Testperson positionierte sich im blauen Zylinder und richtete sich an der Kugel, der weißen Linie und dem zweiten Zylinder aus.
2. Nach Knopfdruck durch die Testperson begann die Kugel sich zu bewegen. Die Testperson folgte ihr in korrektem Abstand auf eine Kreisbahn.
3. Nachdem die Kugel zum Stillstand gekommen war gab die Testperson über den Spielecontroller Rückmeldung, ob sie auf einer Links- oder Rechtskurve gegangen war.
4. Die nächste Startposition wurde eingeblendet und der Testlauf wiederholte sich.

3.4 Datenerfassung & Analyse

Während des Testlaufs wurden die Positionen der Testperson und der Kugel sowie die Kopfausrichtung der Testperson mit ca. 60 Hz erfasst und gespeichert. Pro Testlauf wurden außerdem die Startposition, die Geschwindigkeit und die Rückmeldung gespeichert.

Die gespeicherten Daten wurden in MATLAB analysiert. In Analysen zum Abstand zwischen Kugel und Testperson, Genauigkeit zum vorgegebenen Pfad und tatsächlich gegangener Geschwindigkeit wurde eine grobe qualitative Abschätzung vorgenommen und aufgrund dessen eine Testperson vom Versuch ausgeschlossen.

Nach jedem Testlauf gaben die Testpersonen Rückmeldung, ob sie den Eindruck hatten, auf einer Links- oder Rechtskurve gegangen zu sein. Für die Rückmeldungen der Testpersonen wurde die Korrektheit zur Kurvenbiegung ermittelt und für Rechtskurven invertiert. Nun wurde der Anteil zu den verschiedenen Kurvenbiegungen ermittelt, semantisch nun dem Anteil der Rückmeldungen einer Linkskurve entsprechend. Für die entstandenen Daten wurde eine angenäherte psychometrische Funktion bestimmt. Anhand dieser kumulativen Normalverteilung wurden der Punkt der subjektiven Gleichheit (*Point of Subjective Equality, PSE*) sowie die psychometrische Unterscheidungsgrenze bestimmt, nach Geschwindigkeit gruppiert und vergleichend dargestellt. Für jede Geschwindigkeit wurde außerdem die Standardfehlerrate ermittelt und ebenfalls dargestellt.

4 Ergebnisse und Diskussion

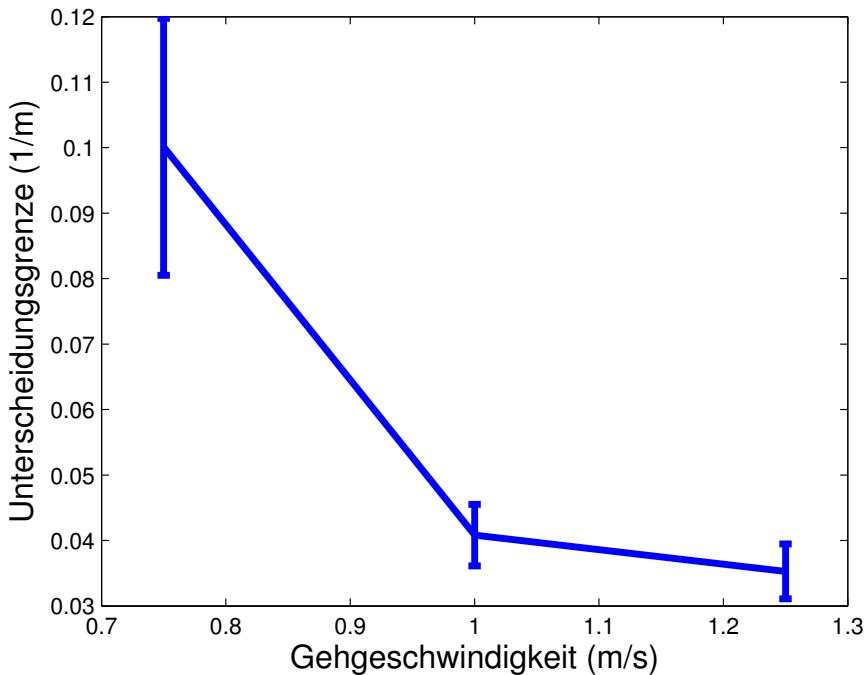


Abbildung 5: Unterscheidungsgrenzen

In der abschließenden Analyse der Unterscheidungsgrenzen (vgl. Abb. 5) zeigte sich, dass die Testpersonen Kurven bei langsamem Gehen erst ab einer größeren Biegung erkannten als bei schnellem Gehen. Die durchschnittliche Unterscheidungsgrenze lag für 0,75 m/s bei einer Kurvenbiegung von $0,098\text{m}^{-1}$, was einem Radius der Kreisbahn von 10,235m entspricht. Für 1,00m/s und 1,25m/s fanden wir Werte von $0,041\text{m}^{-1}$ und $0,035\text{m}^{-1}$ (Radien 24,450m bzw. 28,490m). Wir führten eine ANOVA der Testergebnisse durch und fanden eine Signifikanz von 0,004 ($F=14,225$, $\alpha=0,05$).

Eine unterschiedliche Gehgeschwindigkeit könnte also eine Erklärung für die starke Varianz der Ergebnisse von Kallie et al. (2007) im Vergleich zu Cratty & Williams (1966) und Steinicke et al. (2009) sein. Die Ergebnisse aller drei Versuche liegen innerhalb der von uns gefundenen Grenzen. Da Menschen im Allgemeinen dazu tendieren, in virtuellen Welten langsamer als gewöhnlich zu gehen, erscheinen hier größere anwendbare Kurvenbiegungen durchaus plausibel. Dari-

ber hinaus vermuten wir eine geringere Unterscheidungsgrenze für eine virtuelle Welt mit stärkeren optischen Reizen, da die visuelle Wahrnehmung die propriozeptive Wahrnehmung tendenziell überwiegt (Wallach 1987). Ein möglicher Grund für die geringere Erkennbarkeit bei langsameren Geschwindigkeiten wird in der Überdeckung der seitlichen Umleitung durch die durch kürzere Schritte häufigere Wankbewegung des Gehens vermutet, die detaillierte Analyse steht noch aus.

Die Analyse des Punkts der subjektiven Gleichheit zeigt dementsprechende Ergebnisse. Zwar liegen die Mittelwerte für alle drei Geschwindigkeiten ungefähr bei Null, allerdings sinkt die Standardabweichung von ungefähr $\pm 0,15\text{m}^{-1}$ für 0,75m/s auf ca. $\pm 0,03\text{m}^{-1}$ für 1,00 bzw. 1,25m/s. Das lässt ebenfalls auf eine verminderte Unterscheidungsgenauigkeit für langsame Geschwindigkeiten schließen.

Literaturverzeichnis

Cratty, B. J.; Williams, H. G. (1966): Perceptual Thresholds of Non-Visual Locomotion, Part II. California University, Los Angeles.

Engel, D.; Curio, C.; Tcheang, L.; Mohler, B.; Bülthoff, H. (2008): A psychophysically calibrated controller for navigating through large environments in a limited free-walking space. Max-Planck-Institut für biologische Kybernetik, Tübingen.

Kallie, C. S.; Schrater, P. R.; Legge, G. E. (2007): Variability in Stepping Direction Explains the Veering Behavior of Blind Walkers. Journal of Experimental Psychology: Human Perception and Performance, Vol. 33, 2007, Seiten 183-200.

Nitzsche, N; Hanebeck, U. D.; Schmidt, G. (2004): Motion Compression for Telepresent Walking in Large Target Environments. Presence, Vol. 13, 2004, Seiten 44-60.

Razzaque, S.; Kohn, Z.; Whitton, M.C. (2001): Redirected Walking. Eurographics 2001.

Steinicke, F.; Bruder, G.; Hinrichs, K.; Jerald, J.; Frenz, H.; Lappe, M. (2009): Real Walking through Virtual Environments by Redirection Techniques. Journal of Virtual Reality and Broadcasting, Vol. 6, 2009.

Wallach, H. (1987): Perceiving a stable environment when one moves. Annual Review of Psychology 38, 1–29.

Echtzeitcomputergrafik - Interaktive virtuelle Welten im Entertainmentbereich

Achim Lang

Medien- und Kommunikationsinformatik, Hochschule Reutlingen
Achim.Lang@Student.Reutlingen-University.DE

Abstract:

Mittels aufwändiger Methoden und spezieller Software ist es heutzutage möglich, Bilder auf den Bildschirm oder gar auf Kinoleinwände zu zaubern, die von real aufgenommenen Bildern nicht mehr zu unterscheiden sind. Diese im Pre-Rendering Verfahren generierten Filme werden im Vorhinein mithilfe großer Render-Farmen berechnet und anschließend mit leistungsschwächeren Computern abgespielt. Interaktivität ist dabei nicht möglich oder wird zumindest sehr stark eingeschränkt. Videospiele im Gegensatz setzen hohe Interaktivität voraus und müssen gleichzeitig auf handelsüblichen Computern und Konsolen gespielt werden können. Ein Teil der Computergrafik befasst sich mit der Optimierung bekannter und neuer Methoden zur Darstellung realistischer virtueller Welten in Echtzeit. Dabei spielen sowohl Hardware- als auch Softwareoptimierung eine große Rolle.

1 Definitionen

1.1 Computergrafik

Die Computergrafik befasst sich mit der Erstellung von Bildern mithilfe des Computers. Dazu zählt zum Beispiel, wie komplexe Modelle dreidimensional definiert werden und zweidimensional dargestellt werden können.

1.2 Echtzeit

Der Prozess von Eingabe über Verarbeitung bis hin zur Ausgabe erfordert einen gewissen Zeitrahmen z . Der Begriff „Echtzeit“ beschreibt nicht den Fall, dass z gegen Null strebt, sondern dass z kontinuierlich kleiner als ein vorgegebener Wert bleibt. Bei Computerspielen werden so zum Beispiel 25 Bilder pro Sekunde vorausgesetzt. Liegt die Frame-Anzahl darunter wird dies als „Ruckeln“ des Bildes empfunden. In diesem Fall gilt $z = 1000\text{ms} / 25 = 40\text{ms}$, was wir im Folgenden als *Echtzeitcomputergrafik* ansehen werden.

2 Einführung

2.1 Stellenwert von Videospielen

Videospiele werden schon seit Jahren nicht mehr nur ausschließlich für Computer entwickelt. Eine Vielzahl unterschiedlicher Konsolen sind mittlerweile im Handel erhältlich. Dazu gehören die Xbox 360, Playstation 3, Nintendo Wii und viele mehr. In Bezug auf den Marktanteil der Unterhaltungsindustrie hat die Videospielindustrie die Filmindustrie bereits überholt (Hooker 2008).

Die große Nachfrage an interaktiven Videospielen hat dazu geführt, dass spezielle Computer-Hardware oder kompletten Spielkonsolen entwickelt wurden.

Videospielsoftware wurde und wird auf diese vorhandene Hardware angepasst, um eine optimale Leistungssteigerung zu gewährleisten. Das Zusammenspiel von Hard- und Software kann deshalb als Motor der Spielindustrie betrachtet, da zum einen leistungsstarke Software eine gute Hardware als Basis voraussetzt und zum anderen Hardware ohne passende Software nutzlos ist.

2.2 Interaktive virtuelle Welten

Neben anderen Spiele Genres legen die Entwickler vor allem bei Ego-Shootern, bei denen der Spieler sich durch virtuelle Landschaften bewegen kann, großen Wert auf realistische Darstellung der Umgebung. Das Ziel ist, es dem Spieler zu ermöglichen in virtuelle Welten einzutauchen und alles andere um ihn herum zu vergessen – die so genannte Immersion. Visuelle/akustische Effekte sowie die Echtzeitdarstellung tragen dazu besonders bei. Da der Detailgrad der Realität jedoch ins Unendliche ragt, kann diese nicht 1:1 nachgebaut werden. Überflüssige Informationen, die unnötig Rechenleistung in Anspruch nehmen würden, müssen demnach gefiltert und notwendige Informationen optimiert werden, um die Nutzung der Ressourcen so gering wie möglich zu halten. Das folgende Kapitel beschreibt, welche Möglichkeiten heutige Hardware bietet und wie Software diese effizient nutzt.

3 Leistungssteigerung durch Hardware- und Software-Optimierung

3.1 Hardware-Optimierung

3.1.1 Erhöhung der Taktfrequenz

Wie viele Aufgaben der Prozessor in einer vorgegebenen Zeit bearbeiten kann hängt neben der Befehlsabarbeitung und der Parallelisierung von der

Taktfrequenz des Prozessors ab. Wird die Frequenzzahl erhöht, so werden Prozesse schneller abgearbeitet. Laut Buchheit (2006) sind dieser Leistungssteigerung jedoch physikalische Grenzen gesetzt. Die 3 GHz Grenze ist aufgrund von Stabilitätsproblemen und zu hoher Hitzeentwicklung nur schwer zu überschreiten. Mittlerweile sind deshalb Systeme mit mehreren Prozessorkernen, einer besseren Befehlsabarbeitung (IPC – „Instructions per Cycle“) und einer optimierten Gesamtarchitektur effizientere Lösungen zur Steigerung der Leistung.

3.1.2 Mehrprozessorsysteme

Systeme in denen mehrere Prozessoren verbaut sind und anfangs hauptsächlich als Server verwendet wurden haben aufgrund der physikalischen Taktfrequenzgrenze immer mehr an Bedeutung gewonnen. Zwar resultiert aus der Verdopplung der Prozessoren nicht eine Verdopplung der Leistung, jedoch muss die CPU nicht auf maximaler Taktfrequenz arbeiten und kann somit Energie einsparen wie folgendes Schaubild zeigt.

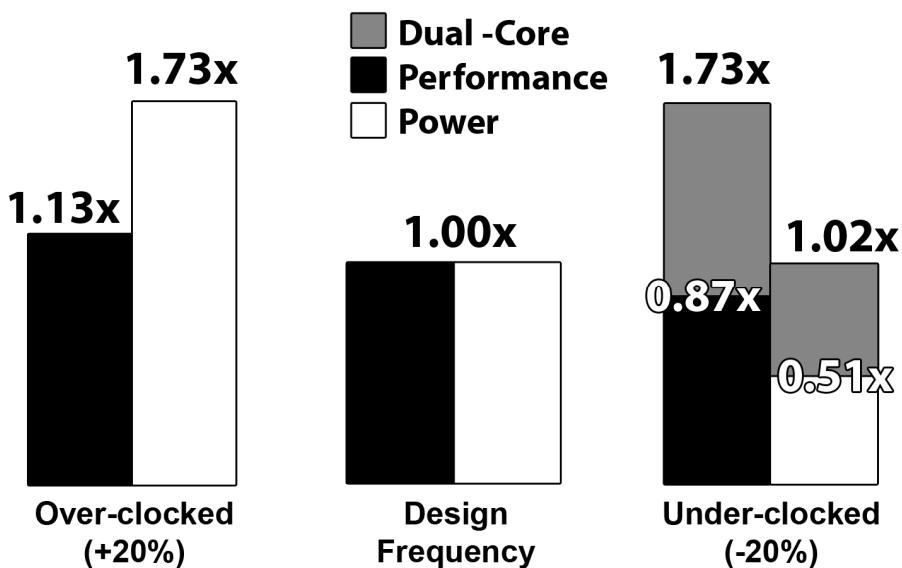


Abbildung 1: Leistung/Energieverbrauch. Links: übertakteter Single-Core. Rechts: Untertaktetes Dual-Core System. Auf den Energieverbrauch bezogen ist zu sehen, dass ein Dual-Core System effizienter arbeitet als ein übertakteter Single-Core (Oram 2010).

Wenn ein Prozessor übertaktet wird, wird zwar die Rechenleistung gesteigert, jedoch steht diese nicht in Relation mit dem Energieverbrauch. Eine Leistungssteigerung von 13% kann zum Beispiel eine Energieverbrauchssteigerung von 73% haben (Oram 2010). Eine effizientere Lösung ist deshalb, die Prozessoren auf ein optimales Leistung/Energieverbrauch – Level anzupassen und sie dann als Multi-Core System zusammenzuschalten. Erwünschter Nebeneffekt dabei ist, dass die Wärmeentwicklung reduziert wird.

3.1.3 Spezielle Befehlssätze: Beispiel SIMD

Eine Komponente der Flynnnschen Klassifikation ist das so genannte SIMD-Prinzip („Single Instruction Multiple Data“). Dieses erlaubt Parallelismus auf einzelnen Prozessoren. Die Idee lässt sich am besten an einem kurzen Beispiel erklären: Möchte man beispielsweise die Helligkeit eines Bildes erhöhen, so muss jeder Pixel neu berechnet werden. Der Befehl „Erhöhe Helligkeit!“ bleibt jedoch immer derselbe. Mit zusätzlichen Befehlssätzen wie SIMD, die von der Hardware unterstützt werden müssen, ist es möglich den Befehl nur ein einziges Mal für mehrere Pixel des Bildes anzuwenden. Die Berechnung dieser Pixel erfolgt dann parallel. Das von Intel entwickelte SSE3 („Streaming SIMD Extensions 3“) erlaubt es beispielsweise mithilfe von zusätzlichen Registern vier Datensätze parallel zu bearbeiten.

3.1.4 GPU

Eine der bedeutendsten Hardwareentwicklungen für die Darstellung von virtuellen Welten ist wohl die „Graphics Processing Unit“. Längst hat sie nicht nur noch die Aufgabe der grafischen Darstellung auf dem Bildschirm, sondern nimmt der CPU ebenso anspruchsvolle 2D- und 3D-Berechnungen ab. Sie besteht aus vielen einzelnen stark vereinfachten Prozessoren, die auf eine spezielle Aufgabe wie zum Beispiel die Vektorberechnung optimiert sind. Laut NVIDIA (2010) können hunderte einzelne Prozessoren in einer GPU verbaut sein, die Befehle der CPU entgegennehmen und parallel abarbeiten. Die Daten werden dabei vom Hauptspeicher in einen lokalen Speicher geladen, verändert und anschließend wieder zurückgegeben. Man spricht dabei von „General Purpose Computation on Graphics Processing Unit“ (GPGPU 2010), da die ursprüngliche Verwendung der GPU nicht mehr im Vordergrund steht. Mithilfe von APIs kann auf die GPU zugegriffen werden um sie für medizinische, wissenschaftliche, mathematische und weitere Zwecke zu verwenden. Des Weiteren verfügen manche Grafikkarten über zusätzliche Physikeinheiten, die beispielsweise Kollisionsberechnungen von Objekten übernehmen können.

3.2 Software-Optimierung

Mit der Fülle der aufgezählten Hardware liegt es in den Händen des Softwareentwicklers, diese effizient zu nutzen. Folgende Möglichkeiten sind Beispiele, die dazu beitragen können.

3.2.1 Optimierung der Algorithmen

An oberster Stelle steht die effiziente Nutzung von Ressourcen anhand optimierter Algorithmen. Redundante Rechenoperationen oder schlecht gesetzte Schleifen können ein Programm ins unermessliche verlangsamen. Bei der Entwicklung von Computerspielen werden deshalb Überwachungsprogramme eingesetzt, die besonders viel Zeit in Anspruch nehmende Aufgaben ermitteln. Die Schwierigkeit besteht dann darin, diese Programmteile dann effizienter zu gestalten – manchmal sogar auf einem niedrigeren Level (Assemblersprachen). Vor der Einführung der Multiprozessorcomputern gab die Taktfrequenz weitestgehend die Geschwindigkeit der Applikationen vor – schlechte oder gar alte Algorithmen waren zwar langsamer, jedoch konnten die Entwickler darauf hoffen ein besseres Ergebnis mit neuen Computern zu erhalten. Laut Buchheit (2006) erlischt dieser positive Effekt sobald eine auf Single-Core geschriebene Anwendung auf einem Multi-Core System ausgeführt wird. Parallelе Software unterscheidet sich von sequentieller Software darin, dass Multi-Core Programme aus einzelnen Teilen (sog. Threads) bestehen, die gleichzeitig ausgeführt werden können und anschließend wieder zusammengeführt werden.

3.2.2 Sinnvolle Parallelisierung

Gerade im Entertainmentbereich werden Konsolen oder Computer an ihre Leistungsgrenzen getrieben. Momentan erhältliche Geräte wie zum Beispiel die Playstation 3 mit eingebautem Cell-Prozessor oder Computer mit eingebautem Intel i7 verfügen über mehrere parallelisierte Einheiten, die bei Spielen bis nahezu 100% ausgelastet werden. Dies ist allerdings nur möglich, wenn die Software einen hohen Grad an Simultanität besitzt. Folgendes Schaubild zeigt Parallelismus bezogen auf den Update- und den Render-Vorgang einer virtuellen Welt. Das letzte der vier aufgezeigten Beispiele soll verdeutlichen, dass die Dauer einzelner Prozesse nicht vorausgesagt werden können und Aufgaben flexibel verteilt werden müssen um den Ressourcen möglichst keinen Leerlaufen zu gewähren.

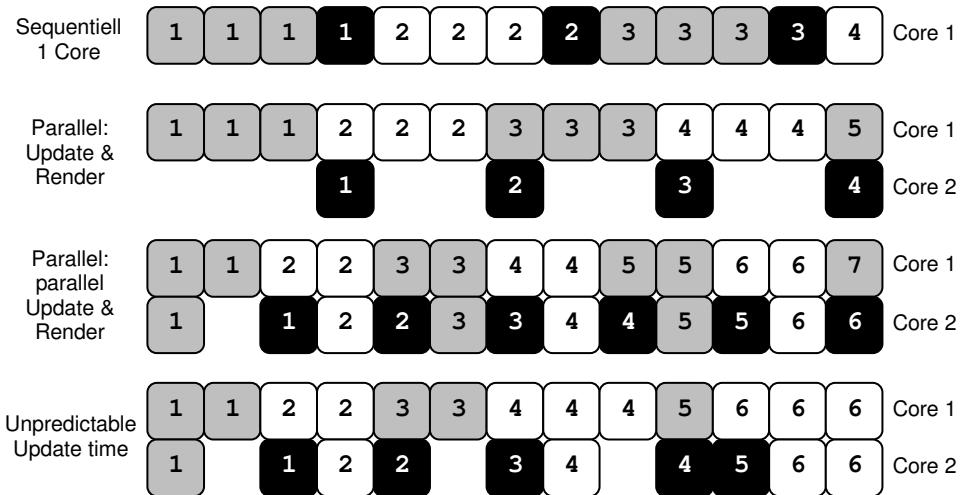


Abbildung 2: Von oben nach unten: Ein Beispiel für sequentielle Update-Vorgänge(weiß/grau) und Render-Vorgänge(schwarz) und drei Beispiele für Parallelismus. Zu beachten sind die unterschiedlichen Anzahlen der gerenderten Bilder. Letztes Beispiel: Unvorhergesehene Zeit für Berechnung des Update-Vorgangs.

Literaturverzeichnis

- Hooker, H. (2008): Packaged Video Games Outsell DVD and Blu-ray in 2008, Gamerev - The Gamer Reviews.
<http://www.thegamereviews.com/story-2842-Packaged-Video-Games-Outsell-DVD-and-Bluray-in-2008.html> (01.04.2010)
- Buchheit, M. (2006): Das Ende der Taktfrequenz.
<http://it-republik.de/dotnet/kolumnen/Das-Ende-der-Taktfrequenz-031331.html> (01.04.2010)
- NVIDIA - Official Website (2010): GPU v. CPU
http://www.nvidia.com/object/nv_ftc_technology.html (01.04.2010)
- GPGPU - Official Website (2010): About GPGPU.org
<http://gpgpu.org/about> (01.04.2010)
- Oram, A. (2010): Foliensatz: Multiprocessing and Parallel Systems
 Sheffield Hallam University

Service-orientierte Architekturen für die Integration von Ingenieurprogrammen im Rahmen der digitalen Produktentwicklung

Thomas Lauria

Institut für Statik und Dynamik der Luft und Raumfahrtkonstruktionen, Universität Stuttgart
Medien- und Kommunikationsinformatik, Hochschule Reutlingen
vincenzo_thomas.luria@student.reutlingen-university.de

Betreuer: Priv.-Doz. Dr.-Ing. Stephan Rudolph, Universität Stuttgart
Prof. Dr.-Ing. Peter Hertkorn, Hochschule Reutlingen

Abstract:

Am Beispiel einer Softwareplattform für die digitale Produktentwicklung wird gezeigt, wie verschiedene Ingenieurprogramme über Webservices miteinander kommunizieren und zusammenarbeiten können. Dabei werden die am häufigsten verwendeten Techniken SOAP und REST miteinander verglichen.

1 Einleitung

1.1 Modellgetriebene Entwicklung

Bei der Produktion von Software wird bereits heute erfolgreich mit Methoden der modellgetriebenen Entwicklung automatisch Programmcode aus zuvor definierten Modellen generiert. Eine mögliche Nutzung der hierbei verwendeten Modellierungssprachen wie UML oder SysML beschränken sich aber nicht nur auf die Domäne der Softwareentwicklung. Hinter der modellgetriebenen Entwicklungen steckt die Idee, aus einfachen Modellen umfangreiche Systeme mittels vorgefertigter Schablonen zu erstellen. Diese Vorgehensweise lässt sich auch auf andere Disziplinen übertragen bzw. der Grundgedanke existiert dort unter etwas konventionelleren Begriffen wie Modul- und Baukastensysteme schon seit längerer Zeit. So untersucht die IILS¹ mbH in Kooperation mit dem ISD² der Universität Stuttgart inwieweit diese Softwareentwicklungstechniken auf der Basis derartiger Modellierungssprachen auch beim Entwurf von komplexen technischen Produkten eingesetzt werden können, die dann als Entwurfssprachen oder Entwurfsgram-

¹ IILS: Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH

² ISD: Institut für Statik und Dynamik der Luft und Raumfahrtkonstruktionen

matiken bezeichnet werden (Kröplin & Rudolph 2005). Zu diesem Zweck wurde das Modellierungswerkzeug Design Compiler 43 entwickelt (IILS 2010).

1.2 Design Compiler 43

Mit dem Design Compiler 43 (im folgenden DC43 genannt) lassen sich technische Systeme wie Satelliten und Flugzeuge durch Entwurfssprachen beschreiben. Die Entwurfssprachen werden dabei so entwickelt, dass bei der Modifikation von Entwurfsparametern nicht nur die von diesen Parametern abhängigen Werte für die Auslegung von Komponenten neu berechnet, sondern möglicherweise auch andere Entwurfsregeln für den Zusammenbau der Komponenten ausgewählt werden.

Dies hat folgenden Vorteil: Ändern sich bei der herkömmlichen Entwicklung eines Flugzeugs zum Beispiel die Anzahl der Sitzreihen, so muss das Flugzeug komplett neu konstruiert werden.

Bei der digitalen Produktentwicklung mit Hilfe des DC 43 wird lediglich im zentralen Entwurf der entsprechende Parameter „Sitzreihen“ geändert. Daraus lassen sich alle weiteren für die Produktion benötigten Modelle ableiten und automatisch generieren. Dies geschieht mit Hilfe weiterer Software, die über Schnittstellen an den DC 43 angebunden werden muss.

Beispielsweise werden in den Entwurfsmodellen des DC 43 spezifische Parameter für physikalische Gleichungen gesetzt, welche zur Lösung an externe Computeralgebrasysteme übergeben werden. Aus den gewonnenen Daten und dem abstrakten Modell lassen sich mit Hilfe des DC 43 fertige CAD Modelle für die Beschreibung der Geometrie oder Finite-Elemente Modelle für Strömungsberechnungen generieren. Hier kommen ebenfalls externe Programme zum Einsatz.



Abbildung 1: Logo des Design Compiler 43 (IILS 2010)

Der DC 43 selbst basiert auf der Open-Source Plattform Eclipse, welche ursprünglich als integrierte Entwicklungsumgebung für die Programmiersprache Java gedacht war. Inzwischen dient der Kern von Eclipse die Rich Client Platform, auch als Ausgangsbasis für die Entwicklung eigener Applikationen. Eclipse bietet eine offene Plug-In Struktur, um flexibel eigene Funktionalitäten hinzuzufügen und damit bestehende Applikationen erweitern zu können. Dieses Konzept wird auch im DC 43 verwendet (Eheim 2009).

Die Schnittstellen zwischen DC 43 und der bereits erwähnten benötigten Drittsoftware sind ebenfalls als Plug-Ins konzipiert. Aktuell gibt es unter anderem Plug-Ins für die Anbindung der folgenden Software:

- Computer Algebra Systemen, insbesondere:
 - Mathematica
 - Matlab
 - Maple
 - Yacas
- CATIA V5: CAD Software der Firma Dassault Systèmes
- Microsoft Excel

1.1 Aufgabenstellung

Momentan sind diese Plug-Ins so konzipiert, dass der DC 43 und die Drittsoftware auf dem gleichen Computer ausgeführt werden (Abbildung 2). Dementsprechend muss neben dem DC 43 auf jeder Arbeitsstation ebenfalls die benötigte Drittsoftware vorhanden sein.

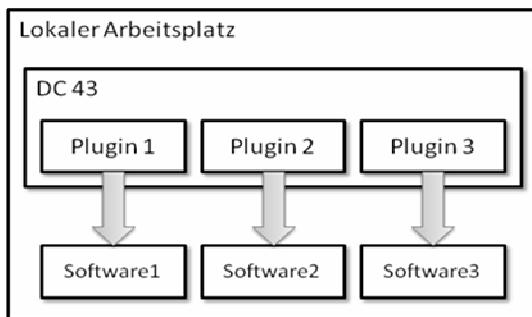


Abbildung 2: DC 43 und Drittsoftware gemeinsam auf einem Arbeitsplatz

Aus diesem Sachverhalt ergibt sich die Aufgabenstellung welche es im Rahmen der wissenschaftlichen Vertiefung zu bearbeiten gilt: Es soll evaluiert werden, in wie weit sich eine Service-orientierte Architektur (SOA) für die Anbindung der Software an den DC 43 umsetzen lässt. Dies bedeutet, dass die benötigte Drittsoftware zentral vorgehalten und durch den DC 43 über ein Netzwerk benutzt werden kann (Abbildung 3).

Aus dieser geänderten, verteilten Struktur ergeben sich mehrere Vorteile:

- Spezielle Hardwareanforderungen der Drittsoftware ziehen keine Upgrades der Arbeitsstationen nach sich
- Der Installationsaufwand wird minimiert, auf einer Arbeitsstation muss lediglich der DC 43 installiert werden
- Rechenintensive Aufgaben können auf separater, entsprechend ausgerüsteter Hardware abgearbeitet werden
- Die Anbindung des DC 43 an die Drittsoftware wird durch eine einheitliche Schnittstelle ermöglicht

Der DC 43 tritt somit als Client auf, während die Drittsoftware auf einem Server als Service angeboten wird.

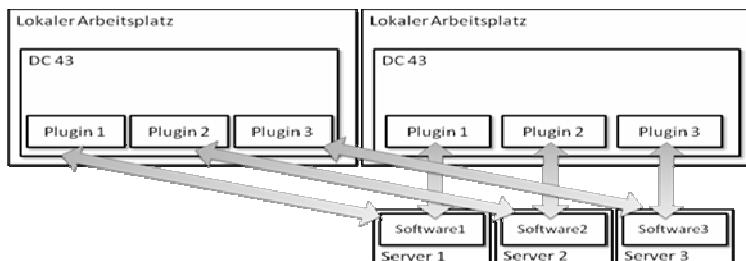


Abbildung 3: Mehrere DC 43 Instanzen greifen auf die Server zu

In der Anfangsphase des Projektes ergab sich eine weitere Anforderung: Die Architektur soll so umgesetzt werden, dass eine technische Entkopplung zwischen Client- und Serverseite stattfindet. Zwangsläufig kommt auf der Clientseite bedingt durch den DC 43 Java zum Einsatz. Die Umsetzung der Serverseite soll davon unabhängig erfolgen, so dass diese mit anderen Programmiersprachen umgesetzt werden kann. Dies impliziert die Kommunikation der Komponenten in einem unabhängigen Datenformat, wobei sich hierfür XML¹ hervorragend eignet.

¹ XML: Extensible Markup Language – erweiterbare Auszeichnungssprache

2 Umsetzung als Web-Service

Java bietet mit RMI¹ und R-OSGi² eigene Lösungen für die Umsetzung einer SOA. Mit RMI können Methoden transparent über ein Netzwerk aufgerufen und Java Objekte übertragen werden. Dabei verbirgt RMI dem Programmierer die Details der Netzwerkverbindung. Ähnlich verhält es sich bei R-OSGi. Grundle-gend beschreibt OSGi den leichtgewichtigen Aufbau einer Software aus einzelnen Modulen. Die R-OSGi Middleware erweitert beliebige OSGi – Implementierun-gen um einen transparenten, verteilten Zugriff. Das dem DC 43 zugrunde liegende Eclipse basiert ebenfalls auf OSGi, was für eine Umsetzung mit R-OSGi sprechen würde. Allerdings steht dies im Widerspruch zu der geforderten technischen Unabhängigkeit von Client und Server, da Lösungen mit RMI wie auch OSGi in Java zu implementieren sind (Abts 2007; Vandenhouten & Kistel 2009).

Mit XML als unabhängigem Datenformat liegt der Schluss nahe die Schnittstelle zwischen DC 43 und der restlichen Software als Web-Service umzusetzen. Ein Web-Service stellt eine URL³ zur Verfügung, über die Daten im XML Format ausgetauscht werden. Die meisten XML Web-Services basieren auf SOAP⁴.

SOAP regelt dabei nicht nur wie die reinen Nutzdaten per XML codiert, sondern auch wie Funktionen zur Steuerung des Web-Services in XML codiert werden. So kann eine URL mehrere Funktionen bereitstellen, aus der XML-Nachricht geht dann hervor welche Funktion auf dem Server ausgeführt werden soll. Gleiches gilt für Antwort an den Client: die XML-Nachricht muss Informationen über Erfolg oder Fehler der Funktionsausführung beinhalten. Durch seine Komplexität gilt SOAP in der Umsetzung allerdings als sehr kompliziert.

Eine bezüglich bestimmter Anwendungsbereiche wesentlich einfachere und elegantere Lösung zu SOAP stellt REST⁵ dar. REST bezeichnet einen Architekturstil von Web-Services der erstmals 2000 in der Dissertation von Roy Fielding (Fielding 2000) auftaucht. REST baut ebenfalls auf HTTP⁶ als Übertragungsprotokoll auf, benötigt aber im Gegensatz zu SOAP kein eigens definiertes Transportprotokoll für die Steuerung der Web-Services, es werden lediglich die Nutzdaten übertragen. Nach REST kann jede Ressource mit einer URL adressiert werden. Für das Auslesen und Manipulieren der zugrundeliegenden

¹ RMI: Remote Method Invocation – entfernter Methoden Aufruf

² R-OSGi: Remote – OSGi; OSGi stand früher für Open Services Gateway Initiative, siehe www.osgi.org

³ URL: Uniform Resource Locator, z.B.: <http://www.infoinside.reutlingen-university.de>

⁴ Simple Object Access Protocol: Protokoll zum einfachen Zugriff auf Objekte

⁵ Representational State Transfer:

⁶ HTTP: Hypertext Transfer Protocol

den Daten baut REST auf die bereits durch HTTP definierten Methoden GET, POST, PUT und DELETE. Mit diesen vier Methoden lassen sich Ressourcen auslesen, anlegen, bearbeiten und löschen. Weiter ist für das Arbeiten mit Ressourcen durch HTTP bereits eine ausreichende Anzahl an Status- und Rückmeldungen definiert. So zum Beispiel der Status 200 im Erfolgsfall, oder der Status 404 wenn eine Ressource unter der angegebenen URL nicht gefunden werden kann. Da bei REST die bereits durch HTTP definierten Methoden und Statusmeldungen benutzt werden, muss keine weitere Transportschicht dazwischen geschoben werden, wie dies bei SOAP der Fall ist. Dementsprechend einfacher ist es, eine REST-basierte Software zu realisieren.

Die Tatsache, dass der Service in Form von Ressourcen umgesetzt wird, bedingt ein Umdenken beim Entwurf einer Software. Ein einfaches Beispiel soll dies verdeutlichen: Ein Fluglinie stellt einen SOAP Web-Service für Flugbuchungen unter der folgenden Adresse bereit:

<http://service/Fluege>

Für die Benutzung der Schnittstelle mit SOAP muss eine entsprechend formulierte XML-Nachricht an die obige Schnittstelle gesendet werden. Die XML – Daten beinhalten die auszuführende Methode z.B.: „BucheFlug“ sowie die Flugnummer und eine Gästenummer. Die Daten werden per HTTP- Methode POST übertragen, die Schnittstelle antwortet mit dem HTTP-Status 200 und einer entsprechenden formulierten XML-Nachricht. Im Fehlerfall würde die Schnittstelle ebenfalls mit Status 200 antworten, der Fehler wäre in den XML- Antwortdaten beschrieben.

Bei REST hingegen wird diese Flugbuchung als Ressource betrachtet und bekommt eine eigene URL:

<http://service/Flug123/Buchung>

Wobei „123“ bereits die Flugnummer ist. Für die Erstellung einer Buchung reicht es aus, per HTTP- POST-Methode die Gästenummer an die obige URL zu senden. Eine weitere Bezeichnung der gewünschten Methode in den Daten fällt weg, da diese bereits über die URL definiert ist. Der Server antwortet lediglich mit einem HTTP Status z.B.: 201 und der neuen URL für diese Buchung. Das Generieren und Versenden einer Antwort in XML fällt weg. Im Falle eines Fehlers müsste dieser nicht als XML übermittelt werden, sondern es wird lediglich ein entsprechender HTTP Status Code ausgespielt (Richardson & Ruby 2007).

3 Umsetzung mit REST

Von vornherein war klar, dass der zeitliche Rahmen der wissenschaftlichen Vertiefung nur eine exemplarische Umsetzung der SOA eines Plug-Ins zulässt. Bereits die Einarbeitung in den DC 43, die Programmiersprache Java und die Plug-In Anbindung nahm einen Großteil der Zeit in Anspruch.

Die Auswahl des Plug-Ins, welches im Rahmen des Projektes umgesetzt werden sollte, ergab sich aus einer aktuellen Problemstellung:

Die Arbeitsplatzrechner wurden auf moderne 64-Bit Systeme umgestellt. Dies hatte zur Folge, dass die Anbindung der Computer Algebra Software Matlab an den DC 43 auf diesen Arbeitsplätzen nicht mehr funktionierte. Dieses Problem wird durch die Umsetzung der SOA hinfällig: die Matlab Software wird auf einer separaten 32-Bit Maschine installiert und die Anbindung an den DC 43 auf dem 64-Bit System wird über die neue REST Schnittstelle möglich.

Für die Umsetzung der Architektur nach REST wurde auf bestehende Bibliotheken zurückgegriffen. So empfehlen die Autoren von Richardson & Ruby 07 das Java Framework RESTlet, welches seit 2005 eine bewährte und verlässliche Software ist, um REST-konforme Systeme zu implementieren. Aus diesen Gründen kommt im DC 43 ebenfalls RESTlet zum Einsatz (Restlet 2010).

Eine weitere externe Bibliothek wird bei der Umwandlung der internen Daten in das für den Transport benötigte XML verwendet. Für diesen Schritt der Serialisierung wird die Software XStream verwendet. Im Gegensatz zu vergleichbaren Produkten stellt XStream keine strukturellen Voraussetzungen an die zu serialisierenden Datenstrukturen in Java. Somit mussten keine Änderungen an den bestehenden Datenobjekten und damit an bestehendem Programmcode durchgeführt werden.

Mit der erfolgreichen Anbindung der Computer Algebra Systeme konnte gezeigt werden, dass das Konzept einer SOA für die Anbindung der Drittsoftware an den DC 43 funktioniert und praktikabel ist.

4 Fazit

Die wissenschaftliche Vertiefung am Institut für Statik und Dynamik der Luft- und Raumfahrtkonstruktionen (ISD) der Universität Stuttgart war in mehrererlei Hinsicht erfolgreich. Ebenfalls bedankt sich der Autor bei der IILS mbH für die vertrauensvolle Zusammenarbeit und die zur Verfügungstellung wesentlicher Informationen.

Zudem hatte der Autor bis zu diesem Zeitpunkt wenig Erfahrung mit der Programmiersprache Java und der Entwicklungsumgebung Eclipse. In diesen Bereichen konnte praktisches Wissen angeeignet werden. Weiter sind die Themen modellgetriebene Entwicklung, REST und Ressourcen orientierte Architekturen Teil der Master-Thesis des Autors, womit die wissenschaftliche Vertiefung eine optimale Vorausarbeit für die Abschlussarbeit darstellt.

Literaturverzeichnis

Kröplin; Rudolph (2005): Entwurfsgrammatiken – ein Paradigmenwechsel?, Der Prüfingenieur. S. 34-43.

Richardson, L.; Ruby, S. (2007): Web Services mit REST. Köln: O'Reilly Verlag, ISBN: 978-3-89721-727-0

Eheim, M. (2009): Erweiterung regelbasierter Transformationsschemata zur automatisierten CFD-Simulation von Rennwagen der Formula Student. Diplomarbeit Universität Stuttgart

IILS mbH (2010): Webseite Entwurfscompiler 43
<http://www.iils.de/> (01.04.2010)

Abts, D. (2007): Masterkurs Client/Server- Programmierung mit Java, 2. Auflage. Wiesbaden: Vieweg Verlag, ISBN: 978-3-8348-0322-1

Vandenhouten, R.; Kistel, T. (2009): Aus der Entfernung – Verteilte Dienste mit R-OSGI, , iX Magazin für Professionelle Informationstechnik. Hannover: Heise Zeitschriften Verlag GmbH & Co. KG. S. 142

Fielding, R. (2000): Architectural Styles and the Design of Network-based Software Architectures. Dissertation, University of California, Irvine

RESTlet (2010): The leading RESTful Web framework for Java
<http://www.restlet.org> (01.04.2010)

XStream (2010): XStream is a simple library to serialize objects to XML and back again. <http://xstream.codehaus.org/> (01.04.2010)

Web 3D - Anwendungsbereiche und Technologien

Armin Wälder, Ulf Maier

Medien- und Kommunikationsinformatik, Hochschule Reutlingen

armin.waelder@student.reutlingen-university.de

ulf.maier@student.reutlingen-university.de

Betreuer: Prof. Dr. Uwe Kloos, Hochschule Reutlingen

Abstract:

Dieser Artikel beschreibt die Arbeitsergebnisse des wissenschaftlichen Vertiefungsprojekts zu Web 3D. Der Fokus liegt dabei auf den Ergebnissen der Evaluation browserbasierter 3D-Lösungen zur Umsetzung und Darstellung von dreidimensionalen Inhalten direkt im Webbrowser. In diesem Zusammenhang werden die 3D-Lösungen 3D Web, O3D, Unity, VRML sowie X3D bewertet. Aus der Analyse der unterschiedlichen Anwendungsbereiche werden einige exemplarische Gebiete und Applikationen des Web 3D vorgestellt. Zusätzlich zu den theoretischen Untersuchungen wurde die Applikation „global.NEWS“ mit der 3D-Lösung Unity umgesetzt, welche ebenfalls in diesem Artikel erläutert wird.

1 Einleitung

Die Darstellung von Internetseiten bzw. deren Inhalten entwickelt sich stetig weiter. Die ursprünglichen browserbasierten Visualisierungsmöglichkeiten von Information beschränkten sich im Wesentlichen auf Bilder und Texte. Im Laufe der Jahre erweiterte sich das Spektrum. Multimediale Inhalte wie Videos oder Flash-Animationen kamen hinzu. Der letzte große Entwicklungsschritt des World Wide Webs war die Integration sozialer Aspekte und Technologien wie Rich Internet Applications im Rahmen des Web 2.0 in den letzten Jahren.

Die zunehmende Leistungsfähigkeit heutiger Server und Client-PCs bietet die Basis für weiterführende Möglichkeiten. Der Ansatz des Web 3D ist eine interaktive und qualitativ hochwertige dreidimensionale Darstellung bzw. die Nutzung von 3D-Anwendungen direkt im Browser. Konsequent und sinnvoll eingesetzt, bietet diese Technologie das Potential die Darstellung sowie die Interaktionsmöglichkeiten von browserbasierten Internetanwendungen zu erweitern und zu verbessern.

Dieser Artikel fasst die Untersuchungs- und Arbeitsergebnisse des Wissenschaftlichen Vertiefungsprojekts zu Web 3D an der Hochschule Reutlingen zusammen. Für eine ausführliche Dokumentation inklusive detaillierter Testtabellen ist Wälder & Maier (2010) zu studieren.

2 Web 3D

2.1 Vorteile und Probleme durch Web 3D

Beim Einsatz von 3D-Applikationen im Web entstehen zahlreiche Vorteile, aber auch einige Probleme.

Die Verbreitung einer Applikation über das Internet kann sehr viele Nutzer erreichen und muss nur zentral auf einem Server gehostet und gepflegt werden. Der Anwender benötigt neben dem PlugIn keine Installationen bei gleichzeitiger Verfüg- und Nutzbarkeit vieler unterschiedlicher Anwendungen. Die einzelnen weiteren Vorteile ergeben sich aus der Charakteristik der jeweils konkreten Anwendung.

Demgegenüber steht neben den erhöhten Anforderungen an die Hardware und mangelnder Nutzererfahrung im dreidimensionalen Raum vor allem die vorhandene PlugIn-Problematik. Derzeit konkurrieren viele verschiedene 3D-Lösungen mit jeweils eigenen PlugIns um die Gunst der Nutzer.

2.2 Anwendungen

2.2.1 Anwendungsbereiche

Der Einsatz von Web 3D bietet in vielen unterschiedlichen Anwendungsgebieten einen Mehrwert im Vergleich zu klassischen Darstellungen. Zu den Anwendungsbereichen gehören Produktpräsentationen, Spiele, Simulationen, Lernumgebungen, Architektur, Kollaborationen und Werbung. Umfassendere Ansätze ermöglichen die Verknüpfung einzelner 3D-Anwendungen zu einer großen VR (siehe Abschnitt 5.2.2). Die folgenden Abschnitte stellen einige Beispiele browserbasierter 3D-Applikationen vor.

2.2.2 Kolonisationssimulation WebScylla

Eine 3D-Simulation zur Visualisierung der Entwicklung in einem künstlichen Riff ist „WebScylla“ (siehe Abbildung 1). Für die Schaffung des künstlichen Riffes namens Scylla, wurde ein Kriegsschiff auf dem Meeresgrund versenkt. Zur Beobachtung der Entwicklung innerhalb des Riffes setzte die Universität von Birmingham in Zusammenarbeit mit dem National Marine Aquarium (NMA) die

Applikation Virtual Scylla um. Tauchern bietet sie die Möglichkeit ihre Sichtungen, bspw. die heimischen Tiere, in die Datenbank der Simulation einzutragen. Um die Zugriffszahlen und die Datenbankeinträge zu erhöhen, wurde eine webbasierte Version des Programmes mit der Bezeichnung WebScylla entwickelt. Die wissenschaftlichen Daten bzgl. der Kolonisation des Riffes und der Populationsfortschritte lassen sich anschließend an einem 3D-Modell des Riffes direkt darstellen. WebScylla bietet somit die Möglichkeit komplexe Zusammenhänge, bspw. die zeitliche und räumliche Ausbreitung verschiedener Spezies, verständlich zu visualisieren. (Vgl. Association for Computing Machinery (2009), S.167 ff.)

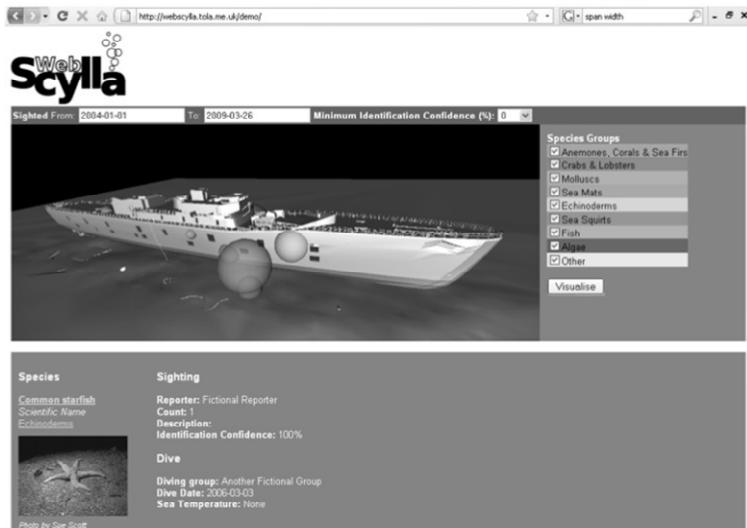


Abbildung 1: Kolonisationssimulation „WebScylla“
(Association for Computing Machinery (2009), S.172)

2.2.3 Ökologische Wirtschaftslernumgebung EnerCities

Eine Lernumgebung für Vorgehensweisen in einer ökologisch nachhaltigen Wirtschaft ist „Enercities“ (Intelligent Energy - Europe 2010). Wie in Abbildung 2 dargestellt, ist es die Aufgabe des Spielers eine Stadt im Stil von SimCity nach diesen Vorgaben zu planen und zu erstellen. Die Anwendung ermöglicht es, auf spielerische Weise das Umweltbewusstsein zu fördern. Diese Teilmenge der Lernumgebungen werden auch „Serious Games“ genannt.



**Abbildung 2: Ökologische Wirtschaftslernumgebung „EnerCities“
(Intelligent Energy - Europe 2010)**

2.2.4 Innenraumkonfigurator SmartDesigner

Ein wichtiges Anwendungsgebiet für Web 3D ist der E-Commerce. Das Beispiel in Abbildung 3 zeigt den Innenraumkonfigurator „SmartDesigner“ (Smart Furniture 2010).

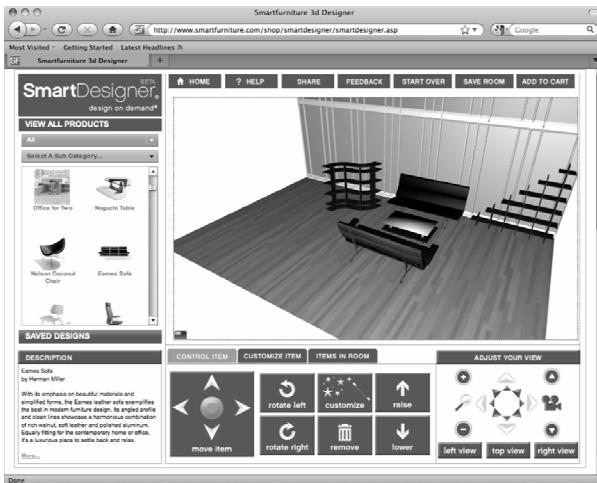


Abbildung 3: Innenraumkonfigurator „SmartDesigner“ (Smart Furniture 2010)

Hierin lässt sich in einem frei definierbaren Raum die Einrichtung der Möbelstücke virtuell vornehmen. Möbel und Möbelkonstellationen können vor

dem Kauf in ihrer Raumwirkung getestet und anschließend bei Bedarf im integrierten Shop direkt erworben werden.

2.2.5 Golfspiel Tiger Woods PGA Tour Online

Ein Spiel, welches zeigt, dass auch aktuelle Computergrafik bzw. aktuelle Computerspiele browserbasiert umgesetzt werden können, ist in Abbildung 4 zu sehen. Die Golfsimulationen „Tiger Woods PGA Tour Online“ (EA Sports 2010) ist eine Adaption des gleichnamigen vollwertigen Computerspiels. Die Umsetzung durch den globalen Spielehersteller Electronic Arts ist zudem ein Indikator für die zukünftige Bedeutung von browserbasierten 3D-Spielen.

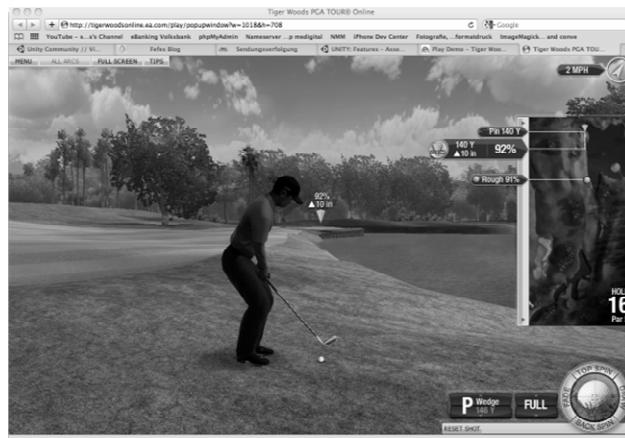


Abbildung 4: Golfspiel „Tiger Woods PGA Tour Online“ (EA Sports 2010)

3 3D-Lösungen

3.1 Bestandteile einer 3D-Lösung

Der Begriff „3D-Lösung“ definiert die Summe aller technischen Maßnahmen eines Ansatzes, welche für eine browserbasierte dreidimensionale Darstellung notwendig sind. Wie in Abbildung 5 dargestellt, besteht eine 3D-Lösung aus einem Webbrowser mit installiertem PlugIn der 3D-Lösung, den 3D-Objekten sowie der zugehörigen Weltenbeschreibung und Skripte für Interaktionen und Verhalten. Darüber hinaus sind Dokumentationen und evtl. vorhandene Anwendungsentwicklerwerkzeuge Bestandteile der Lösung. Die alleinige Betrachtung eines Standards wie X3D wäre somit noch keine Evaluation im Sinne einer gesamten 3D-Lösung. Zusätzlich zum Dateiaustauschformat sind die oben

beschriebenen Elemente zu untersuchen. Die nachfolgenden Unterabschnitte stellen die Bewertungsergebnisse der relevanten 3D-Lösungen vor.



Abbildung 5: Bestandteile einer 3D-Lösung

3.2 3D Web

Die 3D-Lösung „3D Web“ (Lateral Visions 2010c) bietet umfangreiche funktionale Möglichkeiten wie Echtzeit-Reflexionen, Partikeleffekte oder 2D-Textkästen. Hervorzuheben ist der Ansatz eines dreidimensionalen Webs, welches verschiedene Virtuelle Welten sowie 2D-Webcontent ohne Bruch der Immersion verbindet. Realisiert wird dies durch wechselseitige 3D-Hyperlinks in Form von Weltenportalen (siehe Abbildung 6) oder in die VR integrierte Webseiten.



Abbildung 6: Weltenportale zu Verknüpfung einzelner VRs
(The Linnean Society of London 2010)

In Bezug auf Nutzungsplattformen sowie Browser-PlugIns kann 3D Web ebenfalls überzeugen. Alle relevanten Browser sowie die wichtigsten

Betriebssysteme werden unterstützt. Das PlugIn macht keine Probleme und stellt die aufwändigen Virtuellen Welten mit guter Performance und ausreichend flüssig dar. Die Lösung macht insgesamt einen sehr ausgereiften Eindruck. Die 3D-Lösungsentwickler entwickeln darüber hinaus die Lösung stetig weiter. Den vielen positiven Argumenten steht jedoch eine nur mittelmäßige Dokumentation gegenüber. Die Programmierschnittstellen sind zwar umfangreich und detailliert beschrieben, es fehlen allerdings Beispiele oder Tutorials um den Einstieg für Anwendungsentwickler zu erleichtern (Lateral Visions 2010a). Weiterhin ist das Interesse an der 3D-Lösung trotz vieler lokaler Projekte noch sehr gering, was bspw. am offiziellen Forum ersichtlich ist. Durch ein in der Entwicklung befindliches visuelles Werkzeug für Anwendungsentwickler hat die 3D-Lösung gute Chancen auf ein wachsendes Interesse sowie eine zukünftig höhere Verbreitung.

3.3 O3D

Die 3D-Lösung „O3D“ (Google 2010b) von Google befindet sich noch in einem frühen Entwicklungsstadium. Dennoch bietet sie bereits einen umfangreichen Funktionalitätsumfang, bspw. Partikeleffekte, Echtzeit-Reflektionen oder die Nutzung eigener Shader. Durch ihre Konzeption als Low-Level-Lösung lassen sich zudem viele ergänzende Funktionen wie Physik per JavaScript direkt als Zusatz implementieren.

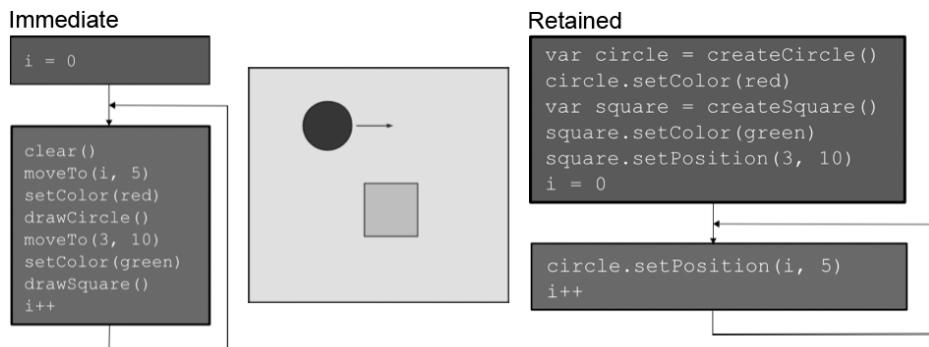


Abbildung 7: Vergleich zwischen Immediate-Modus und Retained-Modus anhand eines Beispiels (Kokkevis & Tavares (2009), S. 7-9)

O3D ist eine tiefansetzende API, welche einen Retained-Modus (Behalten-Modus) nutzt. Im Vergleich zum Immediate-Modus, in welchem bei jedem Frame alle Objekte neu gezeichnet werden, werden bei O3D Informationen über die Objekte zwischen den Frames behalten (siehe Abbildung 7). Dadurch ist die

Ausführung wesentlich effizienter und die 3D-Applikation performanter. (Vgl. Kokkevis & Tavares (2009), S. 7-9)

Es gibt neben der detaillierten API-Spezifikation viele Entwicklerhilfen wie Beispiele, Demos, Tutorials oder Dokumentationen (Google 2010a). Die vorhandene sowie aktive Anwendungsentwickler-Community bietet weitere Anwendungsbeispiele und Hilfestellungen an. Die 3D-Daten müssen für die Nutzung in den meisten Fällen über den Zwischenschritt COLLADA in JSON umgewandelt werden, wodurch die meisten 3D-Applikation unterstützt werden. Alle notwendigen Informationen sind in einem übersichtlichen Webauftritt gebündelt.

Die Anwendungen laufen bereits stabil, allerdings gibt es noch Probleme bei der Lauffähigkeit des PlugIns unter Mac OS X. Weiterhin ist die Linux-Installation noch eine Übergangslösung. Ansonsten ist das PlugIn benutzerfreundlich und beinhaltet die optional nutzbare V8-JavaScript Engine, welche bereits im Webbrowser Chrome ihre Dienste verrichtet. Einzig der mitinstallierte und nicht deaktivierbare Google-Updater trübt evtl. den Betrieb bei vorsichtigen Nutzern. Dem Google-typischen Sammeln von Nutzungsstatistiken kann allerdings explizit bei der Installation widersprochen werden. Da die 3D-Lösung noch nicht fertiggestellt ist, ergeben sich neben den PlugIn-Problemen noch weitere hieraus resultierende Defizite. So gibt es bisher noch kein unterstützendes dediziertes Anwendungsentwicklungswerkzeug, was einen sehr hohen Aufwand bei der Erstellung von 3D-Applikationen bedingt. Daneben sind noch Probleme mit einigen Grafikbeschleunigern bekannt.

Trotz des noch jungen Alters der 3D-Lösung, gibt es bereits einige beeindruckende Umsetzungen. Die IT-Branchengröße Google verleiht O3D zusätzlich Gewicht. Entsprechend hoch war das Interesse bei der ersten Veröffentlichung im April 2009. Google sucht darüber hinaus die offene Konversation mit anderen Entwicklern und ist bestrebt mit O3D einen offenen Webstandard für 3D-Inhalte zu etablieren. Entsprechend stark schreitet die Entwicklung voran. Es werden regelmäßig Bugfixes & Erweiterungen veröffentlicht. Insgesamt beschränkt sich das Interesse derzeit aber noch im Wesentlichen auf Anwendungsentwickler. Bei einer finalen Veröffentlichung und einer gelungenen Öffentlichkeitsarbeit kann O3D aber zukünftig sein vorhandenes Potential entfalten.

3.4 Unity

Die 3D-Lösung „Unity“ (Unity 2010b) konnte in der Evaluation überzeugen und ging als beste Lösung hervor. Das Herzstück der Spieleentwicklungsumgebung ist

das gleichnamige und sehr gute Anwendungsentwicklerwerkzeug (siehe Abbildung 8). Es unterstützt Anwendungsentwickler bei der Erstellung Virtueller Welten und bietet neben einem sehr breiten Funktionsspektrum viel Komfort. Bspw. müssen 3D-Daten oder Texturen nur einmalig importiert werden, die Aktualisierung des Inhalts bei geänderten Dateien wird ebenso automatisch wie die Generierung von Mip- oder BumpMaps vollzogen. Beim Datenimport unterstützt Unity die gängigen 3D-Editoren. Weitere relevante Eigenschaften sind eine Echtzeitvorschau und eine integrierte Abwärtskompatibilität für die erstellten Applikationen. Ein einmal erzeugtes Projekt lässt sich neben dem Export als Webanwendung auch auf weiteren Plattformen veröffentlichen. Aus technischer Sicht ist Unity auf dem aktuellen Stand computergrafischer Verfahren und nutzt auch junge Technologien und Verfahren wie Ageia PhysX oder Screen Space Ambient Occlusion. Ein einfaches PlugIn ist für die meisten System-Browser-Konstellationen verfügbar. Durch die eigene JavaScript-Engine und die Qualitätsanpassungen laufen die Anwendungen stets performant.



Abbildung 8: Entwicklungsumgebung Unity

Neben der Werkzeugunterstützung erfahren Anwendungsentwickler viele verschiedene Hilfestellungen von Dokumentationen über Beispiele bis hin zu Videotutorials. Gesammelt sind diese Informationen auf der ebenfalls sehr guten sowie umfangreichen, informativen und übersichtlich gestalteten offiziellen Webseite (Unity 2010a). Hier hebt sich Unity deutlich von allen evaluierten Konkurrenten ab. Zusätzlich ist eine große Anwendungsentwickler-Community aktiv. Aus Interessensperspektive sprechen weitere Gründe wie der jahrelange Einsatz von großen und kleinen Anwendungsentwicklerfirmen für die 3D-Lösung.

Es sind zudem viele, meist qualitativ hochwertige, Anwendungen verfügbar. Regelmäßige und sehr häufige Weiterentwicklungen sowie ein derzeit starkes Wachstum der Entwicklerfirma unterstreichen die Zukunftsfähigkeit der Lösung. Das Interesse an Unity und damit dessen Relevanz stiegen noch einmal deutlich durch die Veröffentlichung einer kostenfreien Version. Unity zeigt, was technisch in Webbrowsern möglich ist und hat bei anhaltend steigender Verbreitung und umfangreicher Öffentlichkeitsarbeit gute Chancen sich gegen die alternativen vorhandenen und in der Entwicklung befindlichen 3D-Lösungen weiterhin stark zu positionieren.

3.5 VRML

„VRML“ (Web3D Consortium 2010a) ist mit der ersten Veröffentlichung im Jahr 1995 die älteste 3D-Lösung der Evaluation. In Kombination mit der Tatsache, dass X3D bereits seit 2001 als offizieller Nachfolger gilt (siehe Abschnitt 3.6), bietet die Virtual Reality Modeling Language demzufolge die technischen Möglichkeiten aus dieser Entwicklungszeitspanne. Heutige Standards wie Reflexionen oder Partikeleffekte sind folglich nicht vorhanden. Auch der Umgang mit aktuellen medialen Dateiformaten wie MP3 oder die Integration von 2D-Webcontent wird nicht unterstützt. Weiterhin wird die Lösung nicht mehr gepflegt oder weiterentwickelt, womit sie heute eigentlich obsolet ist. Dies ist auch an den vielen nicht mehr verfügbaren Projektwebseiten erkennbar. Das Interesse sank über die letzten Jahre stetig.

Dennoch bietet VRML viele Möglichkeiten wie Kamerafahrten oder die Nutzung von Levels of Detail. Erweiterungen sind über das External Authoring Interface (EAI) möglich und werden sehr häufig eingesetzt. Durch seinen Status als ISO-Standard ist VRML in den meisten 3D-Applikationen als Exportformat nutzbar. Neben den Spezifikationen (Web3D Consortium 2010b) sind Dokumentationen und Hilfen sowie die Community noch für Hilfestellungen erreichbar. Als sehr gut ist die Unterstützung von Systemen und Browern zu bewerten. Schlecht ist dagegen die Unübersichtlichkeit der PlugIns, welche die Spezifikation darüber hinaus unterschiedlich interpretieren. Die PlugIn-Vielfalt ist für Nutzer nicht nur aufgrund fehlender Gesamtübersichten unüberschaubar und anwenderunfreundlich, zusätzlich erfordern verschiedene 3D-Applikation zwingend verschiedene VRML-PlugIns. So kann es vorkommen, dass für eine 3D-Applikation das vorhandene PlugIn deinstalliert und ein neues installiert werden muss. Dies ist ein wesentlicher Grund, weshalb VRML bei der Breite der Internetnutzer und Anwendungsentwickler nur mäßig erfolgreich war. Dennoch ist VRML im direkten Vergleich mit dessen Nachfolger X3D die häufiger genutzte 3D-Lösung. Bill McCloskey, ein ehemaliger Mitarbeiter der VRML-

Unterstützer-Firma SGI, bezeichnet die Entwicklung von VRML im Jahr 2000 als eine Geschichte von „Selbstüberschätzung, Sabotage, Egoismus und Missmanagement“ (McCloskey 2000).

Abbildung 9 zeigt auf der linken Seite eine klassische bzw. optisch repräsentative VRML-Applikation als Computerspiel. Daneben zeigt das rechte Beispiel einer kompletten Stadionpräsentation, bei welcher die Funktionalität von VRML durch Extensions erweitert wurde, dass mit der 3D-Lösung durchaus auch hochwertige Umsetzungen möglich sind.

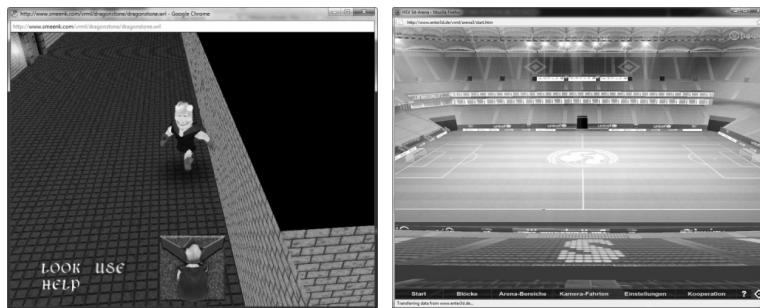


Abbildung 9: VRML-Beispielanwendungen „Dragonstone“ (Smeenk 2010) und „HSV HSH Nordbank-Arena“ (HSV-Arena 2010)

3.6 X3D

Die 3D-Lösung „X3D“ (Web3D Consortium 2010c) bietet im Vergleich zu ihrem direkten Vorgänger VRML (siehe Abschnitt 3.5) einen erweiterten Funktionalitätsumfang, bspw. Partikeleffekte, MP3-Audointegration, ein Physikmodel, Lightmapping oder eine festlegbare Renderingreihenfolge. Weitergehende Funktionen wie Reflexionen und Schattenwurf sind nur über Erweiterungen realisierbar. Die 3D-Datenunterstützung und die Dokumentation der Spezifikation (Web3D Consortium 2010d) sind wie die Systemunterstützung ebenfalls umfangreich. Darüber hinaus wurde von den Entwicklern zusätzlich die Laufzeitarchitektur spezifiziert. Hier wurde aus den VRML-PlugIn-Inkonsistenzen gelernt. Allerdings gibt es konsortiumsbedingt wiederum eine Vielzahl unterschiedlicher Plugins, mit gleichen Problematiken in Bezug auf die Anwenderfreundlichkeit, wie bereits bei VRML geschildert. Dies ist ein wesentlicher Grund für die geringe Verbreitung von X3D. Bei den verfügbaren browserbasierten Anwendungen spielt es praktisch keine Rolle. Soll ein ISO-Standard genutzt werden, ist bei fast allen Projekten noch VRML im Einsatz. Dessen Interesse ist auch bei den Internetnutzern im direkten Vergleich immer

noch höher.

Der zweite Grund für die niedrige Verbreitung ist in der mangelhaften Öffentlichkeitsarbeit zu sehen. In diesem Zusammenhang ist vor allem der Internetauftritt des Web 3D-Konsortiums zu nennen. Trotz oder gerade wegen seiner Vielzahl an namhaften Mitgliedern ist die Webseite sehr unübersichtlich und ungepflegt. Unverständlich ist neben der wenig ansprechenden Gestaltung der Seite, dass der eigentlich umfangreiche Inhalt sehr viele Fehler, bspw. Doppelteinträge oder tote Links aufweist. Hier mangelt es an der notwendigen Konsolidierung zusammengetragener Informationen der Konsortiumsmitglieder. Zusammenfassend wäre eine inhaltliche und optische Neugestaltung des Internetauftritts vor allem für die Einstiegsfreundlichkeit von Anwendungsentwicklern und Nutzern sowie für die generelle Interessenssteigerung von großem Vorteil.

3.7 Weitere 3D-Lösungen

Neben den in den vorherigen Abschnitten analysierten 3D-Lösungen gibt es noch eine Reihe alternativer Ansätze, welche sich teilweise noch in der Entwicklung befinden. Sie werden an dieser Stelle der Vollständigkeit halber samt ihrer offiziellen Internetpräsenz genannt:

- 3DMLW (<http://www.3dmlw.com/>)
- Ajax3d (<http://sourceforge.net/projects/ajax3d/>)
- Blink 3D / inDuality (<http://www.pelicancrossing.com/index.htm>)
- Canvas 3D (<https://wiki.mozilla.org/Canvas:3D>)
- Director (<https://www.adobe.com/products/director/>)
- Java 3D (<https://java3d.dev.java.net/>)
- WebGL (<http://www.khronos.org/webgl/>)
- WireFusion (<http://www.demicron.com/wirefusion/>)
- XML3D (<http://xml3d.com/>)

4 Umsetzung einer Web 3D-Applikation

Zusätzlich zu den theoretischen Untersuchungen wurde die browserbasierte 3D-Applikation „global.NEWS“ (siehe Abbildung 10) mit der 3D-Lösung Unity implementiert. Die Anwendung liest verschiedene vordefinierte RSS-Feeds aus

und visualisiert deren Inhalte mittels einer dreidimensionalen Weltkugel. Der Nutzer kann sich die News sowohl in einem automatischen Modus abspielen lassen als auch den Globus manuell nach Neuigkeiten absuchen.

Für die Entwicklung wurde die 3D-Lösung Unity gewählt, da sie als Testsieger aus der Evaluation hervorging. Die 3D-Lösung beinhaltet zusätzlich eine integrierte Entwicklungsumgebung für Windows und Mac, welche die Umsetzung unterstützt. Durch das Deployment als Webanwendung und den kostenlosen Unity Player kann die Applikation auf allen unterstützten Browsern und Systemen betrieben werden.

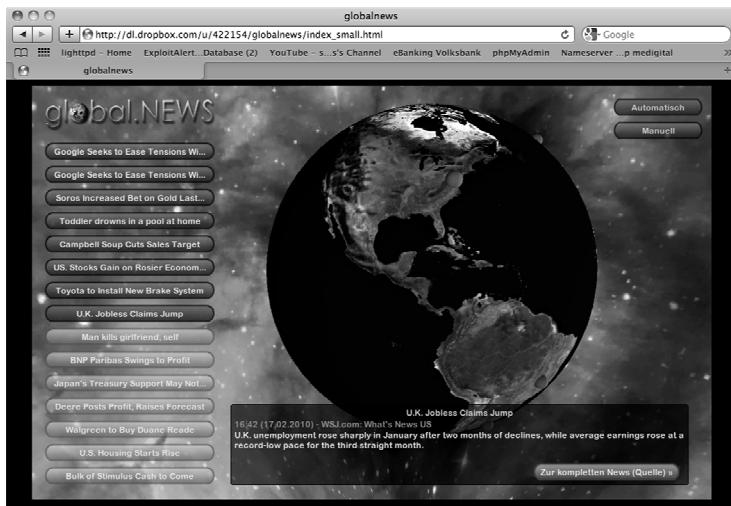


Abbildung 10: Finale 3D-Applikation „global.NEWS“ im Webbrower

5 Fazit

5.1 Zusammenfassung

5.1.1 Anwendungsgebiete

Das dreidimensionale Web wird das klassische 2D-Web nach Meinung der Autoren und heutigem Kenntnisstand nicht ersetzen. Informationen lassen sich nach wie vor über ein 2D-Interface schneller finden, bspw. ein Suchbegriff bei Google und der anschließende Besuch einer Nachrichtenseite. Vielmehr bietet Web 3D die Möglichkeit, das vorhandene Web auf vielfältige Arten zu erweitern.

Neben prädestinierten Anwendungsgebieten wie Produktpräsentationen oder

Computerspiele, sind ebenso webbasierte dreidimensionale Simulationen, Lernumgebungen, Anwendungen zur Architektur und Kollaboration sowie werbetreibende Maßnahmen möglich. Umfassendere Ansätze ermöglichen zudem im komplexesten Anwendungsfall den Aufbau eines parallel existierenden 3D Webs als VR, welche viele kleine Virtuelle Welten zu einer großen Anwendung verknüpft. Viele existierende Web 3D-Anwendungen sind allerdings noch Demonstrationen zur Vermittlung der jeweiligen 3D-Lösungsfunktionalität.

5.1.2 Ergebnisse der 3D-Lösungsevaluation

Zusammenfassend lässt sich aus der Evaluation der 3D-Lösungen ermitteln, dass eine Vielzahl konkurrierender Angebote um die Vorherrschaft im dreidimensionalen Web kämpft. Die offiziellen ISO-Standards in diesem Bereich, VRML und dessen Nachfolger X3D, enttäuschen allerdings und werden von kommerziellen Herstellern deutlich geschlagen. Dies zeigt sich am Interesse der Anwendungsentwickler und der Nutzer sowie den vorhandenen Anwendungen. Es macht den Anschein, als ob in den Konsortien nicht die Kräfte gebündelt werden, sondern stattdessen kleinste gemeinsame Nenner gefunden werden müssen. Die kommerziellen Hersteller wie O3D verkaufen zudem ihre Produkte durch gute Öffentlichkeitsarbeit und umfangreiche Entwicklerunterstützung deutlich besser. Die nach der Evaluation beste 3D-Lösung, speziell mit Vorteilen in diesen Bereichen, ist Unity.

Viele IT-Größen wie Google, Mozilla oder die W3C Group engagieren sich eigenständig oder in Konsortien im Bereich Web 3D. Dies zeigt die aktuelle und zukünftige Relevanz an der Thematik bzw. Technologie. Gleichzeitig sind allerdings der Markt und die Nutzungsbereitschaft der Anwender zu klein für derart viele konkurrierende 3D-Lösungen. Der Hauptnachteil von Web 3D ist derzeit die Vielzahl an unterschiedlich benötigten PlugIns, bedingt durch die vielen einzelnen 3D-Lösungen. Welche Lösung ihre konkurrierenden Ansätze mittel- und langfristig verdrängen wird, ist zum jetzigen Zeitpunkt noch völlig offen. 3D Web bietet interessante Ansätze, O3D ist noch in einer frühen Entwicklungsphase und Unity ist ursprünglich zur Spieleentwicklung konzipiert. Trotz seiner bisher bescheidenen Verbreitung sollte auch der ISO-Standard X3D noch nicht abgeschrieben werden. Neben den ausführlich getesteten Lösungen, gibt es noch eine Reihe kleinerer Nischenprodukte oder in der Entwicklung befindlicher Ansätze. Ein Beispiel ist die geplante native Integration der Lösung WebGL u.a. in den Browser Firefox (siehe hierzu auch Abschnitt 5.2.1). Auch die künftigen Entscheidungen von Microsoft, welche sich in der Thematik Web 3D bisher zurückhalten, werden Gewicht haben. Ebenso gibt es noch keine Antwort auf die Frage, ob sich eine Lösung etabliert oder es eine Koexistenz verschiedener

3D-Lösungen geben wird. Dies wird in erster Linie von der nativen Browserintegration, der Öffentlichkeitsarbeit der Entwickler sowie den mit einer 3D-Lösung umgesetzten Anwendungen abhängen.

5.1.3 Web 3D = Web 3.0?

Web 3D kann als das zukünftige Web 3.0 oder einen Teil dessen Technologie betrachtet werden. Inwiefern dies zutreffend sein wird, hängt in erster Linie von der Nutzerakzeptanz sowie der Bereitschaft der Webentwickler ab, entsprechende 3D-Lösungen zu nutzen bzw. einzusetzen. Das Potential der Technologie ist gegeben und an vielen Lösungen wird fleißig entwickelt. Generell gibt es allerdings keine verbindliche Definition des Begriffes „Web 3.0“.

5.1.4 Praktische Umsetzung

Die praktische Umsetzung des Projektes „global.NEWS“ konnte die vorausgehende Evaluation und deren Ergebnisse bzgl. der 3D-Lösung Unity verifizieren. Sowohl die gleichnamige Entwicklungsumgebung als auch die weit reichenden Hilfen und Anleitungen waren im Praxiseinsatz hilfreich. Trotz dieser klaren Vorteile gegenüber der Konkurrenz traten auch mit Unity größere und kleinere Probleme auf, bspw. wurden Abweichungen von der JavaScript-Spezifikation nicht dokumentiert.

5.2 Ausblick

5.2.1 Natives Web 3D

Das mittelfristige Ziel zur Umgehung der PlugIn-Problematik ist die native Integration der 3D-Funktionalität direkt in den Browser. Dies macht den Einsatz von PlugIns überflüssig. Mehrere Browserhersteller haben dies bereits in ihren Testreleases umgesetzt, sodass WebGL in Firefox, Safari, Opera sowie Chrome zu Testzwecken schon nativ integriert ist (Vgl. Khronos Group 2010). Außerdem plant Google die native O3D-Integration in Chrome (Vgl. CNET 2009). Neben einer größeren Zielgruppe entfallen dann die Installationshürden zumindest für diese PlugIns.

5.2.2 Verknüpfung verschiedener 3D-Lösungen

Wie bereits erläutert, bieten umfassendere Ansätze die Möglichkeit eines großen dreidimensionalen Webs. Einen ersten Schritt hierbei geht die in Abschnitt 3.2 evaluierte 3D-Lösung „3D Web“ durch ihre mit Weltenportalen verknüpften VRs. Für zukünftige Releases ist zudem die Verknüpfung mit Anwendungen geplant, welche mit konkurrierenden Lösungen wie VRML oder X3D erstellt wurden

(Vgl. Lateral Visions 2010b). Ein „wandern zwischen den Welten“ soll so realisiert werden.

5.2.3 Mobiles Web 3D

Generell folgen die grafischen Darstellungsmöglichkeiten auf mobilen Geräten denen der Desktop-Darstellung im Abstand von wenigen Jahren, womit Web 3D mittelfristig auch im mobilen Sektor relevant werden wird.

Literaturverzeichnis

- Association for Computing Machinery (2009): Proceedings Web3D 2009. 14th International Conference on Web3D Technology. ACM, 2009.
<http://portal.acm.org/portal.cfm>
- CNET (2009): Google building 3D hardware boost into Chrome.
http://news.cnet.com/8301-17939_109-10293207-2.html (23.03.2010)
- EA Sports (2010): Tiger Woods PGA Tour Online.
<http://tigerwoodsonline.ea.com/> (23.03.2010)
- Google (2010a): O3D API. <http://code.google.com/apis/o3d/> (23.03.2010)
- Google (2010b): O3D Projektseite. <http://code.google.com/p/o3d/> (23.03.2010b)
- HSV-Arena GmbH & CO. KG (2010): Die Arena in 3D. http://www.hsv-hshnordbank-arena.de/de/die_arena/die_arena_in_3d.html (23.03.2010)
- Intelligent Energy – Europe (2010): EnerCities. <http://www.enercities.eu/> (23.03.2010)
- Khronos Group (2010): WebGL. <http://www.khronos.org/webgl/> (23.03.2010)
- Kokkevis, V.; Tavares, G. (2009): O3D: Adding Interactive 3D Content to your site. Google IO 09. http://dl.google.com/io/2009/pres/W_0300_AddingInteractive3DContenttoyourSite.pdf (23.03.2010)
- Lateral Visions (2010a): 3D Web Entwicklerseite. <http://dev.lv3dweb.net/> (23.03.2010)
- Lateral Visions (2010b): 3D Web Features.
http://www.lateralvisions.com/3D_Web_Technology/Features.aspx
(23.03.2010)

- Lateral Visions (2010c): 3D Web Übersichtsseite.
http://www.lateralvisions.com/3D_Web_Technology/Overview.aspx
(23.03.2010)
- McCloskey, B. (2000): The Rise and Fall of VRML. Part 1:
<http://www.clickz.com/821941> & Part 2: <http://www.clickz.com/822061>
(23.03.2010)
- Smart Furniture (2010): SmartDesigner.
<http://www.smartfurniture.com/shop/smardesigner/smardesigner.asp>
(23.03.2010)
- Smeenk, R. (2010): VRML Beispiel Dragonstone.
<http://www.smeenk.com/vrml/dragonstone/dragonstone.wrl> (23.03.2010)
- The Linnean Society of London (2010): Linnean Society 3D Website.
<http://3d.linnean.org.uk> (23.03.2010)
- Unity (2010a): Unity Entwicklerunterstützung. <http://unity3d.com/support/>
(23.03.2010)
- Unity (2010b): Unity Produktseite. <http://unity3d.com/> (23.03.2010b)
- Wälder, A.; Maier, U. (2010): Web 3D - Evaluation browserbasierter 3D-Lösungen. Ausarbeitung zur Wissenschaftlichen Vertiefung. Hochschule Reutlingen, 2010.
- Web3D Consortium (2010a): Archivierte VRML Webseite.
<http://www.web3d.org/x3d/vrml/> (23.03.2010)
- Web3D Consortium (2010b): VRML Spezifikation.
<http://www.web3d.org/x3d/specifications/vrml/> (23.03.2010)
- Web3D Consortium (2010c): X3D Produktseite. <http://www.web3d.org/x3d/>
(23.03.2010)
- Web3D Consortium (2010d): X3D Spezifikation.
<http://www.web3d.org/x3d/specifications/> (23.03.2010)

Adaptive Systeme - Gestaltung und Einsatz

Janina Bierkandt, Michael Bierkandt

Medien- und Kommunikationsinformatik, Hochschule Reutlingen

Janina.Bierkandt@Reutlingen-University.DE

Michael.Bierkandt@Student.Reutlingen-University.DE

Abstract:

Adaptive Systeme analysieren ihre Umgebung und Benutzer und passen sich an sie an. Sie bieten ein großes Potenzial für die Vereinfachung und Verbesserung interaktiver Systeme, dies wird allerdings derzeit noch nicht vollständig ausgeschöpft. Die Systematik adaptiver Systeme wird vorgestellt und ihr Aufbau erläutert. Zusätzlich wird betrachtet, welche Faktoren hinsichtlich Usability eine Rolle spielen. Beispiele zeigen mögliche Adaptionen auf.

1 Einleitung

1.1 Definition

Der Begriff Universal Design bzw. Universal Usability und das damit verbundene Ziel, ein Produkt so zu gestalten, dass es von möglichst vielen unterschiedlichen Nutzern in jeglichen Situationen benutzt werden kann (Mace 1998), gewinnt in den letzten Jahren immer mehr an Bedeutung (Shneidermann 2003). Nach Mace ist die Nutzung für alle Nutzer ohne Anpassung oder Modifizierung möglich.

Im Bereich der Mensch-Maschine Schnittstellen erscheint dieses auf Grund der hohen Komplexität und des Funktionsumfangs meist unmöglich. Bei der Gestaltung und Festlegung von Richtlinien und Normen wird deshalb auf Möglichkeiten der Anpassung - Personalisierung bzw. Individualisierung - gesetzt (Shneidermann 2003; ISO 9241). Die Anpassung soll auf die aktuelle Situation und die Nutzer abgestimmt sein (Kühme 1993) und die zunehmende Erfahrung der Nutzer berücksichtigen (Dieterich et al. 1993).

Erfolgt die Anpassung durch ein explizites Eingreifen von außen, spricht man von adaptierbaren Systemen (Oppermann 1994; Nitschke et al. 2001), passt sich das System selbstständig an veränderte Bedienungen an, handelt es sich um ein adaptives System (Oppermann 1994; Kobsa et al 2001). In Lernumgebungen spricht man auch von Makro- und Mikro-Adaption (Leutner 2002), da davon ausgegangen wird, dass die automatische Adaption kleinere Anpassungen vornimmt und größere explizit durch den Nutzer angestoßen werden (Petko & Reusser 2006).

Langley (1997) geht in seiner Definition von adaptiven Systemen zudem von lernenden Systemen aus, die sich selbst regulieren. Die zugrunde liegenden Modelle werden ständig erweitert bzw. angepasst. Durch diese Rückkopplung verbessern sich die Möglichkeiten und die Qualität der Anpassung. Wahlster (2003) und Dieterich et al. (1993) ziehen einen Vergleich zur künstlichen Intelligenz.

Jameson (2003) bringt in seiner Definition beide Arten von adaptiven Systemen zusammen: „An [...] adaptive system can be defined as an interactive system that adapts its behavior [...] on the basis of processes of [...] acquisition and application that involve some form of learning, inference, or decision making.“ Für die weitere Betrachtung der Gestaltung und des Einsatzes von adaptiven Systemen kommt diese Definition zum Einsatz.

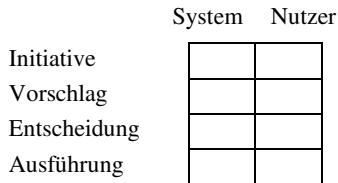
1.2 Klassifizierung

Dieterich et al. (1993) unterscheiden aufbauend auf den Überlegungen von Totterdell & Rautenbach (1990) vier Phasen innerhalb des Adoptions-Prozesses: Initiative, Vorschlag, Entscheidung und Ausführung.

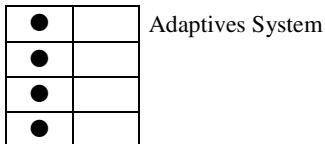
In der ersten Phase, der Initiative, wird entschieden, dass eine Anpassung vorgenommen werden soll. Alternativen zur Anpassung des Systems werden in der Vorschlagsphase erarbeitet und zur Auswahl gestellt. In der dritten Phase, der Entscheidung, wird eine der Alternativen ausgewählt, die letzte Phase führt die ausgewählte Anpassung durch.

Jeder dieser Phasen kann automatisch vom System durchgeführt werden oder manuell durch den Nutzer des Systems. Werden alle Phasen manuell ausgeführt, handelt es sich um ein adaptierbares System, werden im Gegensatz dazu alle Phasen automatisch durchgeführt, spricht man von einem adaptiven System (vergleiche Abschnitt 1.1).

Theoretisch sind 16 Kombinationen möglich (2 Akteure, 4 Phasen), allerdings erscheinen nicht alle sinnvoll. Die Ausführung durch den Nutzer, nach dem ein Vorschlag automatisch generiert wurde, erscheint nicht sinnvoll, unabhängig von der Initiative und Ausführungsphase. Ebenso erscheint es nicht sinnvoll, die Entscheidung für eine Alternative automatisch zu treffen, die Ausführung im Anschluss jedoch manuell vornehmen zu lassen, da für die Entscheidung die Alternativen analysiert und verstanden werden müssen und eine automatische Ausführung dadurch problemlos möglich sein sollte. Durch diesen Ausschluss ergeben sich acht sinnvolle Kombinationen, diese werden nachfolgend dargestellt und benannt. Es wird dazu folgende Matrix verwendet:



Adaptive Systeme



Adaptierbare Systeme

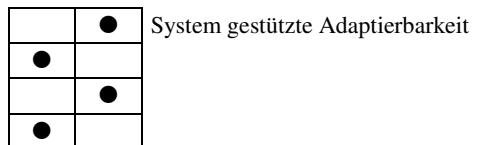
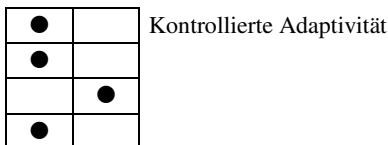
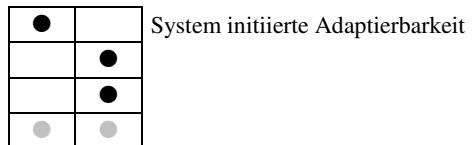
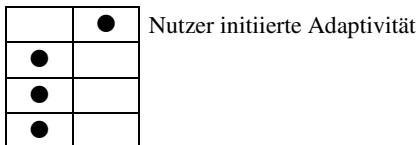
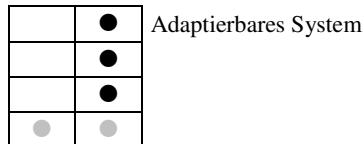


Abbildung 1: Klassifizierung nach Akteuren und Phasen (vgl. Dieterich et al. 1993)

Die Ausprägungen der adaptiven Systeme unterscheiden sich durch den Ansatzpunkt des Nutzers – Initierung der Anpassung (Nutzer initiiert) oder Auswahl einer Anpassung (kontrolliert).

Greift der Nutzer in mehr als einer Phase ein, wird von adaptierbaren Systemen gesprochen, da es sich nicht mehr um eine automatische Anpassung handelt (vergleiche Abschnitt 1.1). Initiiert das System eine Anpassung, wird unabhängig davon, wer die Ausführung schließlich durchführt, von einem System initiierten adaptierbaren System gesprochen. Die Art der Ausführung – durch System oder Nutzer – ist für die Klassifizierung von adaptierbaren Systemen ebenfalls irrelevant. Bei System gestützten adaptiven Systemen entscheidet der Nutzer über den Zeitpunkt und die Art der Anpassung.

Zur weiteren Klassifizierung von adaptiven Systemen wird die Art der Anpassung bzw. die Art der Unterstützung herangezogen, die der Nutzer durch die Anpassung

erfährt. Nach Browne (1990) kann zwischen einer Anpassung der Kommunikation (Anpassung der Gestaltung, ohne Veränderung des Funktionsumfangs) und einer der Funktionalität (Anpassung des Funktionsumfangs) unterschieden werden. In dieses Schema lassen sich adaptive Hilfesysteme, Systeme die Routineaufgaben übernehmen und Systeme die offensichtliche Fehler korrigieren (vergleiche Hoppe & Plötzner 1991) allerdings nur schwer einordnen. Dieterich et al. (1993) wählt deshalb eine Klassifizierung nach generischen Funktionen (Hilfe- und Fehlerbehebungssysteme) und der Interaktion (Präsentation, Funktionsumfang und Vereinfachung). Jameson (2003) unterscheidet vier Typen der Unterstützung:

1. Taking Over Parts of Routine Task
2. Adapting the Interface
3. Giving Advice About System Use
4. Controlling a Dialog

Jameson (2003) betrachtet die Systeme noch stärker in Bezug auf die Art der Unterstützung des Nutzers und umgeht somit scheinbar die Schwierigkeit der klaren Abgrenzung zwischen Funktionalitäts- und Kommunikationsaspekten. Allerdings wird die Abgrenzung von Typ 2 zu Typ 4 dadurch unscharf und ist im Nachhinein nicht vollständig nachvollziehbar. Es ist anzunehmen, dass Jameson in Typ 2 die Anpassung von Funktionalitäten und in Typ 4 die Anpassung der Kommunikationsaspekte zusammenfassen wollte. Für die weitere Betrachtung der Systeme wird dennoch von nur drei unterschiedlichen Typen ausgegangen, zumal sich Kommunikations- und Funktionalitätsaspekte in neueren adaptiven Systemen oft nicht scharf trennen lassen:

1. Teile von Routineaufgaben übernehmen
2. Mensch-Maschine-Schnittstelle anpassen
3. Hilfestellung zur Nutzung des Systems

2 Gestaltung

2.1 Aufbau

Nach Oppermann (1994) lässt sich der Aufbau adaptiver Systeme in drei wesentliche Elemente aufteilen:

1. Daten sammeln (Afferenz), die zur Adaption des Systems notwendig sind

2. Daten auswerten (Inferenz) und eine passende Adaption auswählen
3. System adaptieren (Efferenz)

Diese Aufteilung ist in Abbildung 2 zusammengefasst und wird in den folgenden Abschnitten näher erläutert.

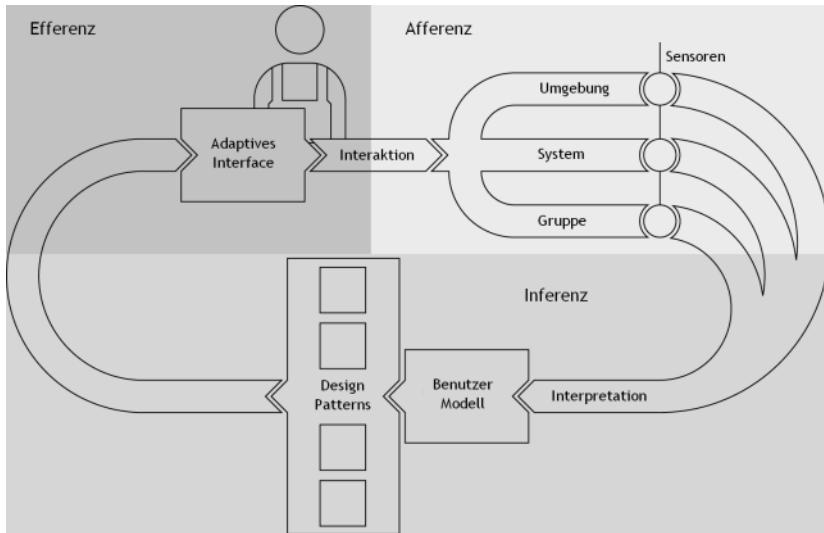


Abbildung 2: Wesentliche Elemente eines adaptiven Systems (nach Bierkandt 2009)

2.1.1 Daten sammeln (Afferenz)

Um sich adaptieren zu können, muss ein System Daten zur Verfügung haben, die Rückschlüsse auf sinnvolle Adaptionen erlauben. Hierbei wird zwischen der impliziten und der expliziten Datensammlung unterschieden.

Implizite Datensammlung

Bei der impliziten Datensammlung trägt das System eigenständig im Hintergrund Daten zusammen, die Benutzer sind nicht direkt beteiligt. Es können vier verschiedene Bereiche charakterisiert werden.

(1) Bei der kontextbezogenen bzw. die Aufgabenmerkmale betreffenden Datensammlung steht der aktuelle Kontext des Nutzers im Vordergrund, das bedeutet das System analysiert, welchen Vorgang der Benutzer momentan durchführt. Es kann beispielsweise betrachtet werden, welche Bildschirm-Maske dem Benutzer momentan angezeigt wird. Generell werden diese Vorgänge als natürlich auftretenden Aktionen bezeichnet (vergleiche beispielsweise Jameson

2001). Es handelt sich demzufolge um Aktionen, die die Benutzer bei der normalen Nutzung des Systems ausführen.

(2) Die Sammlung der Benutzermerkmale konzentriert sich auf die Profile der Benutzer, das heißt es werden benutzerspezifische Informationen gesammelt. Hierzu gehören hauptsächlich Informationen über Aktionen und Funktionen, die ein Benutzer bereits mit dem System durchgeführt hat. Zusätzlich können allgemeine Nutzungsparameter des Systems von Interesse sein, wie zum Beispiel die Auswertung von Serverlog-Dateien. Nach Jameson (2003) können außerdem unabhängige Systeme, zum Beispiel öffentlich zugängliche Quellen wie Internetseiten, zur Datensammlung herangezogen werden.

(3) Die technischen Merkmale betreffen die Mensch-Maschine-Schnittstelle. Je nach System kann der Nutzer mit unterschiedlichen Geräten (PC, Mobiltelefon...) und Benutzeroberflächen (Webbrowser, Anwendung...) agieren. Auch die Anbindung, das heißt welche Bandbreite für die Datenübertragung zur Verfügung steht, kann variieren.

(4) Von den technischen Merkmalen abhängig sind die Umgebungsmerkmale, da je nach System die Sensorik zur Datenerfassung variieren kann. Prinzipiell bietet die Sensorik dem System die Möglichkeit, Informationen über seine Umgebung zu sammeln. Es ist der Einsatz von zahlreichen unterschiedlichen Sensoren möglich, beispielsweise zur:

- Orts- und Positionsbestimmung
- Messung der Umgebungshelligkeit
- Gesichtserkennung
- Emotionserkennung
- Blickrichtungserfassung

Sensoren können entweder kontinuierlich Daten sammeln, beispielsweise in Intervallen oder bei bestimmten Ereignissen.

Vorteil der impliziten Datensammlung ist, dass der Benutzer seine gewohnte Interaktion mit dem System ausführen kann. Dies ist jedoch insofern auch ein Nachteil, da die implizite Datensammlung intransparent für den Benutzer ist (vergleiche Jameson 2003). Da der Benutzer nicht weiß, welche Daten wann und wie gesammelt werden, kann ihn dies verunsichern oder sogar dazu führen, dass er sich in seiner Privatsphäre verletzt fühlt und insgesamt der Benutzung des Systems kritisch gegenübersteht (vergleiche Abschnitt 3.2 und 3.3).

Explizite Datensammlung

Bei der expliziten Datensammlung sind die Nutzer aktiv an der Datensammlung beteiligt. Frühere explizite Ansätze wie beispielsweise von Liang (1987) betrachten das Einbeziehen der Benutzer in den Gestaltungsprozess adaptiver Systeme. Diese Vorgehensweise ist in der Praxis oft nicht praktikabel, da in der Regel mehrere Iterationsphasen notwendig sind und bei einer Vielzahl unterschiedlicher Nutzer nur schwer alle während der Designphase miteinbezogen werden können. Davon abgesehen kann die explizite Datensammlung in zwei Bereiche unterteilt werden.

- (1) Einstellungen, die Benutzer aktiv im System vornehmen. Hierbei kann zwischen Einstellungen unterschieden werden, die Benutzer für sich selbst festlegen und Einstellungen, die für andere Benutzer vorgenommen werden, beispielsweise durch Administratoren in Form von Rechtevergaben.
- (2) Explizite Abfrage von Informationen. Hier können Techniken wie bei klassischen Fragebögen (Multiplechoice bzw. standardisiert) und Bewertungssysteme zum Einsatz kommen, aber auch verpackte oder verschleierte Abfragen zum Beispiel innerhalb beziehungsweise in Form eines Spiels.

Mit der expliziten Datensammlung lassen sich in der Regel die gleichen Daten erfassen wie auch bei der impliziten Datensammlung. Jedoch ist nach Jameson (2003) zu berücksichtigen, dass den Antworten von Benutzern nicht bedingungslos zu vertrauen ist. Beispielsweise könnten Benutzer nicht der Wahrheit entsprechend antworten, weil sie eine Antwort als unangemessen empfinden (Soziale Erwünschtheit). Hier müssen daher die Erkenntnisse und Methoden der Psychologie zur Fragebogengestaltung berücksichtigt werden. Ein weiterer Nachteil ist die oft relativ aufwändige Dateneingabe, abhängig davon welche Informationen und wie viel Daten gesammelt werden. Nach Carroll & Rosson (1989) sind Benutzer eines Systems in der Regel nicht Willens, für (aus ihrer Sicht) nicht zielführende Abfragen größere Anstrengungen zu akzeptieren.

2.1.2 Daten auswerten (Inferenz)

In der Inferenzphase werden die gesammelten Daten verarbeitet und hinsichtlich der Adaption ausgewertet. Mögliche sinnvolle Rückschlüsse sind beispielsweise:

- welche Aktion(en) der Benutzer als nächstes ausführen will
- welchen Wissensstand der Benutzer in Bezug auf das System/Themenfeld hat
- wie aktuelle Einflüsse aus der Umgebung die Interaktion mit dem System beeinträchtigen

- welche Adaptionen technisch gesehen überhaupt möglich, sinnvoll und erlaubt sind

Die Berechnung derartiger Benutzermodelle ist keinesfalls trivial, es können nach Kobsa (2004) unterschiedliche Ansätze verfolgt werden.

Regelbasierte Ansätze verknüpfen Aktionen von Benutzern fest mit bestimmten Rückschlüssen, beispielsweise bestimmte Eigenschaften des Benutzers oder die Einordnung in bestimmte Nutzergruppen, so genannte Stereotypen.

Algorithmen zur Vorhersage von Aktionen wie zum Beispiel Künzer et al. (2004) oder auch Planerkennungsmethoden, versuchen anhand der gesammelten Daten die wahrscheinlichsten nächsten Aktionen von Benutzern zu berechnen.

Probabilistische Verfahren berücksichtigen die Unsicherheit von Schlussfolgerungen und beziehen die Möglichkeit mit ein, dass sich Rückschlüsse aus gewonnenen Daten widersprechen können oder dass mehrere mögliche Rückschlüsse existieren.

Mittels maschinelner Lernverfahren wird versucht, Muster in den gesammelten Daten zu erkennen und darauf aufbauend Rückschlüsse zu ziehen. Dadurch kann sich das Verhalten des Systems im Gegensatz zu rein regelbasierten Ansätzen im Verlaufe der Zeit ändern (vergleiche lernende Systeme Abschnitt 1.1)

Beim kollaborativen Filtern werden andere Benutzer zur Schlussfolgerung herangezogen, die dem aktuellen Benutzer am Ähnlichsten sind. Mit Hilfe von Clusterverfahren werden Stereotype gebildet.

Prinzipiell kann nach Oppermann (1994) zwischen halb-automatischer und automatischer Adaption unterschieden werden. Bei ersterer kann der Benutzer die Adaption beeinflussen oder sogar verhindern.

Neben der Problematik, wie ein System die gesammelten Daten richtig auswerten und interpretieren kann, sind nach Jameson (2001) auch die Berechnungskomplexität in Bezug auf die Systemressourcen und die zur Schlussfolgerung notwendige Zeit zu berücksichtigen. Des Weiteren spielt die Menge der mindestens notwendigen, gesammelten Daten eine Rolle, da insbesondere beim Beginn der Nutzung eines Systems keine oder nur wenig gesammelte Daten vorliegen.

2.1.3 System adaptieren (Efferenz)

Mit Efferenz ist die Umsetzung der Adaption also die Veränderung des Systems gemeint. Möglichkeiten für Adaptionen und Beispiele werden später aufgezeigt.

2.2 Usability von adaptiven Systemen

Zur Gestaltung interaktiver Systeme die ein hohes Maß an Usability und User Experience aufweisen, wird in der ISO Norm 9241-210 der Human-Centered Design Prozess definiert. Dieser Prozess sieht eine starke Einbeziehung der Nutzer in den Gestaltungsprozess vor und weißt auf die unterschiedlichen Bedürfnisse und Anforderungen in bestimmten Situationen hin. Zudem wird auf die Veränderlichkeit dieser Aspekte über Zeiträume und Nutzungssituationen hinweg aufmerksam gemacht (ISO 9241-210).

Adaptive Systeme scheinen hier eindeutige Vorteile gegenüber anderen interaktiven Systemen zu haben, da Nutzer auch nach dem Gestaltungsprozess bzw. der Einführung mit einbezogen werden können und dadurch auf Veränderungen eingegangen werden kann.

Klar wird bei dieser Betrachtung allerdings auch, dass das Verständnis für die Aufgaben, Bedürfnisse und Anforderungen der Nutzer in bestimmten Situationen eine noch größere Rolle spielt, da dies die Grundlage für die Anpassung der Systeme bildet. Mangelndes Verständnis bzw. Missinterpretation wird bereits bei traditionellen interaktiven Systemen als einer der häufigsten und kritischen Fehler bei der Gestaltung dieser identifiziert (ISO 9241-210). Daher werden im Folgenden die sieben Gestaltungsregeln der ISO Norm 9241-110 in Bezug auf adaptive Systeme bewertet.

Adaptive Systeme bieten den Nutzern Möglichkeiten zur Individualisierung bzw. passen sich automatisch an die speziellen Gegebenheiten und Bedürfnisse an. Durch diese Anpassung können adaptive Systeme dazu beitragen, dass Fehleingaben vermieden werden. Zudem können die Hilfestellungen speziell auf die Nutzer abgestimmt werden, so dass eine optimale Unterstützung der Nutzer erreicht wird. Durch die Möglichkeit der Anpassung an die aktuelle Situation und Aufgabe der Nutzer, kann außerdem eine gute Aufgabenangemessenheit erreicht werden. Adaptive Systeme können somit im Rahmen der Definition der ISO Norm 9241 prinzipiell als individualisierbar, fehlertolerant und aufgabenangemessen gestaltbar angesehen werden.

Die Möglichkeit gezielte, auf die Situation und den aktuellen Nutzer abgestimmte Informationen anzubieten, bietet eine ideale Voraussetzung zur Gestaltung von Lernförderlichkeit und Selbstbeschreibungsfähigkeit. Da es sich allerdings um veränderliche Systeme handelt, können vom Nutzer erlernte Abläufe nach einer Anpassung des Systems unter Umständen nicht mehr zum gewünschten Ziel führen. Zudem erscheint es nicht in allen Fällen sinnvoll, den Nutzer explizit und umfassend über vorgenommene Anpassungen zu informieren. Diese Aspekte

machen es notwendig, der Gestaltung von Lernförderlichkeit und Selbstbeschreibungsfähigkeit in adaptiven Systemen besondere Bedeutung zu schenken.

Erwartungskonformität scheint in adaptiven Systemen nur schwer abbildbar zu sein (Jameson 2003), da sich das System je nach aktuellem Kontext unterschiedlich verhalten kann. In lernenden Systemen wird dieser Effekt noch verstärkt, da sich die Regeln, nach denen das System angepasst wird, während der Nutzung verändern können. Ähnlich verhält es sich mit dem Aspekt der Kontrolle, insbesondere wenn Nutzer nicht einbezogen werden. Untersuchungen bestehender Systeme bestätigen diese Annahmen und zeigen in den genannten Aspekten deutliche Schwächen adaptiver Systeme auf (Gajos et al. 2008).

In den letzten Jahren wurden Untersuchungen zur Überprüfung der Auswirkungen dieser Schwächen, die allen adaptiven Systemen gemein sind, durchgeführt. Erste Ergebnisse zeigen geringere Auswirkungen auf Usability und User Experience als erwartet (Paymens et al. 2004). Es stellt sich daher die Frage, ob diese Gestaltungsregeln für adaptive Systeme nicht gelten, die Schwächen durch andere Aspekte ausgeglichen werden oder sich die Nutzer in den letzten Jahren dahingehend verändert haben, dass Kontrollierbarkeit und Erwartungskonformität eine weniger wichtige Rolle spielen. Auf Grund des derzeitigen Forschungsstandes erscheint es ammaßend, etablierte Gestaltungsregeln in Frage zu stellen. Es wird deshalb im weiteren Verlauf davon ausgegangen, dass die Schwächen durch entsprechende Gestaltung ausgeglichen werden können, so dass sie keinen oder nur minimalen Einfluss auf die Nutzer haben.

Zusätzlich zu den Aspekten der ISO-Norm 9241 scheinen für adaptive Systeme weitere Faktoren maßgeblich für eine gute Usability zu sein. Jameson (2003) fasst diese Aspekte in drei weiteren Gestaltungsregeln zusammen: Unaufdringlichkeit, Privatsphäre und Erfahrungstiefe.

(1) Unaufdringlichkeit: Wird der Nutzer aktiv in den Adoptionsprozess oder die Sammlung von Daten eingebunden, kann dies zu einer erhöhten Belastung der Nutzer führen. Auch die Art der Einbindung und die jeweilige Situation in der sich die Nutzer befinden spielt eine Rolle. Beispielsweise können Abfragen vom Nutzer als penetrant und aggressiv empfunden werden, wenn sie sich ständig wiederholen oder flüssige Abläufe unterbrechen.

(2) Privatsphäre: Zur Gestaltung adaptiver Systeme sind persönliche Daten der Nutzer nahezu unabdingbar. Um die Akzeptanz von adaptiven Systemen zu gewährleisten, muss der Privatsphäre der Nutzer daher besonderer Respekt gezollt und dem Nutzer dadurch die Sicherheit gegeben werden, dass seine persönliche Daten keinem Missbrauch unterliegen.

(3) Erfahrungstiefe: Durch die automatische Anpassung der Systeme werden dem Nutzer kürzere, optimierte Bearbeitungsabläufe angeboten. Neben all den Vorteilen die diese Anpassung bietet, verringert sich die Tiefe mit der sich der Nutzer mit dem System bzw. den Informationen beschäftigt. Dies kann unter Umständen negative Auswirkungen auf Lerneffekte für andere Bereiche haben. Zudem kann der Nutzer weniger Alternativen in seine Entscheidung einbeziehen. Es könnte z.B. passieren, dass vom System Alternativen ausgeschlossen werden, die der Nutzer eventuell auf Grund von Faktoren die nicht im System erfasst sind, ausgewählt oder in Betracht gezogen hätte. Dies kann zudem negative Auswirkungen auf die hinterlegten Benutzermodelle haben, da diese für den Nutzer in dieser Situation wichtigen Faktoren nicht aufgenommen werden.

2.3 Gestaltung

2.3.1 Prinzipien

Unabhängige Experten bewerteten im Schnitt nur etwa 7 % der vom System vorgeschlagenen/vorgenommenen Anpassung als sinnvoll (McGrenere et al. 2007). Die Ursachen sind in der Afferenz und der Inferenz zu suchen, bzw. in den dahinterliegenden Modellen.

Genaue Analysen, welche Informationen und Querverbindungen benötigt werden, sind deshalb unabdingbar. Dienst- und Schichtpläne bilden beispielsweise eine gute Grundlage, können aber erst in Kombination mit aktuellen Uhrzeit- und Datumsinformationen sinnvoll genutzt werden. Die Auswahl der Datentypen und Datengrößen sollte insbesondere in lernenden Systemen unter Berücksichtigung zukünftiger Anforderungen betrachtet werden.

In Bezug auf die Datensammlung sollte der Aufwand und die Art der Sammlung näher analysiert werden. Die Performanz bzw. die problemlose effiziente Nutzung des Systems sollte dabei immer im Vordergrund stehen und durch die Datensammlung nicht beeinträchtigt werden. Explizite Datenabfragen dürfen den Nutzer z.B. in kritischen Situationen nicht ablenken. Im Idealfall werden explizite Methoden mit impliziten kombiniert – so kann eine Vielzahl von unterschiedlichen Daten gesammelt und die Eingaben der Nutzer in Bezug zu den implizit gesammelten Informationen bewertet werden.

Beschreibungen über den Zweck der Datensammlung, die Weiterverarbeitung (Speicherung) der Daten und darüber welche Daten erfasst werden können helfen, die Akzeptanz der Nutzer zu steigern. Mechanismen, mit denen die Nutzer entscheiden können, welche Daten erfasst werden dürfen, steigern das Vertrauen der Nutzer, gleichzeitig geben sie ihnen das Gefühl der Kontrolle über das System. Die Informationen und Mechanismen sollten dabei möglichst leicht

verständlich und anwendbar sein, ebenso sollten Änderungen effizient vorgenommen werden können.

Zur Interpretation der gesammelten Informationen muss ein Modell bzw. ein Regelwerk erarbeitet werden, dass die unterschiedlichen Informationen miteinander in Verbindung bringt, so dass geeignete Arten der Anpassung ausgewählt werden können. „Störungen“ der Daten im Vergleich zu bereits gesammelten Referenzdaten sollten durch das Modell genauestens analysiert werden, da durch diese Fehlinterpretationen entstehen können. Insbesondere bei zeitlich befristeten Störungen bietet sich eine aktive Einbeziehung der Nutzer an, die darüber Auskunft geben können ob die Daten zur Interpretation herangezogen werden sollten. Zusätzlich wird den Nutzern so das Gefühl der Kontrolle vermittelt, sie bekommen einen Einblick in die Prozesse der Adaption und können zukünftige Anpassungen besser nachvollziehen.

Bei der Auswahl einer geeigneten Adaption bietet sich ebenfalls die aktive Einbeziehung der Nutzer an. Die Auswahl einer Alternative kann als zusätzliche Information für zukünftige Adoptionsprozesse dienen. Für die Auswahl sollte nur eine begrenzte Anzahl an Alternativen angeboten werden, die für den Nutzer klar voneinander trennbar sind. Gezielte Informationen über die Unterschiede zwischen den Alternativen erleichtern dabei die Auswahl und erhöhen die Akzeptanz. Die Vorauswahl des Systems auf Grundlage der gesammelten Informationen erleichtert dem Nutzer die Auswahl und steigert zudem das Gefühl der Transparenz und der Kontrolle über das System in einem hohen Maße. Werden zu den Alternativen zusätzliche Informationen angeboten, warum diese Art der Anpassung vorgeschlagen wird bzw. auf welchen Informationen die Auswahl beruht, kann dies die Akzeptanz zusätzlich steigern. Zudem können falsche Annahmen (Daten) vom Nutzer identifiziert und im Idealfall eliminiert werden, dies kann die Qualität insbesondere in lernenden Systemen steigern.

Es bleibt dennoch im Einzelfall abzuwagen, ob eine explizite Auswahl durch den Nutzer sinnvoll erscheint, da dieser dadurch in seinem derzeitigen Handeln unterbrochen und somit gezwungen wird, eine Entscheidung zu treffen. Bei kleineren oder häufig auftretenden Anpassungen sollte auf eine explizite Einbeziehung pro Einzelfall verzichtet werden. Darüber hinaus sollte geprüft werden, ob Übertragungen aus Einzelfällen möglich sind oder ob dem Nutzer generische Entscheidungen abverlangt werden können.

Stehen keine Alternativen zur Verfügung oder sind bestimmte Anpassungen notwendig, um den Nutzer bei der Bearbeitung seiner Aufgabe unterstützen zu können, erscheint eine Auswahl unter Einbeziehung der Nutzer trotz der genannten Vorteile nicht sinnvoll.

Bei der Auswahl einer geeigneten Anpassung bzw. den Alternativen, die dem Nutzer zur Auswahl angeboten werden, sollte auch Aspekte der Erfahrungstiefe betrachtet werden – in einigen Fällen erscheint es sinnvoll, auch Anpassungen in Betracht zu ziehen, die in Bezug auf die gesammelten Daten nicht die optimale Lösung darstellen (Jameson 2003).

Bei der Ausführung der Adaption ist darauf zu achten, den Nutzer nicht bei der derzeitigen Nutzung einzuschränken, d.h. die Adaption sollte z.B. nicht dazu führen, dass Eingaben erneut getätigt werden müssen. Ist kein fließender Übergang möglich, sollten die Nutzer über die Adaption informiert werden. Im Idealfall werden dabei zusätzliche Informationen angeboten, die den Nutzern dabei helfen, die Anpassung nachvollziehen zu können. Wurde eine Adaption durchgeführt sollte diese im Idealfall zudem auf Wunsch des Nutzers rückgängig gemacht werden können. In einigen Fällen erscheint es allerdings sinnvoller den Nutzer nicht über die Anpassung zu informieren, beispielsweise wenn mit der Adaption einem Fehlverhalten des Nutzers entgegen gewirkt werden soll.

Zusammenfassend kann festgehalten werden, dass die aktive Einbeziehung von Nutzern und gezielte Informationen in allen Phasen des Adoptionsprozesses die Akzeptanz der Nutzer steigern können. Der Adoptionsprozess sollte den Nutzer in der Ausübung seiner Aufgaben nicht beziehungsweise so wenig wie möglich beeinträchtigen.

2.3.2 Adaptiv versus adaptierbar

Die aktive Einbeziehung der Nutzer wurde als Schlüsselfaktor für die Usability und die User Experience beschrieben. Von diesem Standpunkt aus betrachtet scheinen adaptierbare Systeme besser zur Anpassung geeignet als adaptive Systeme (vergleiche Abschnitt 1.2). Werden alle Phasen des Anpassungsprozesses durch den Nutzer ausgeführt, ist allerdings zu bedenken, dass Nutzer meist nicht ausreichend über die Anpassungsmöglichkeiten informiert sind und sie zudem ihre eigenen spezifischen Präferenzen und Bedürfnisse oft nur schwer fassen können beziehungsweise sich dieser nicht bewusst sind (Kobsa 1994; Rauterberg 1996). Nutzer müssen also zunächst Erfahrungen mit einem System sammeln und über mögliche Anpassungen informiert werden bevor sie Adaption gezielt selbstständig vornehmen können.

Aufbauend auf diesen Erkenntnissen erscheint eine Kombination von adaptiven und adaptierbaren Mechanismen in den meisten Kontexten am besten geeignet. Die verschiedenen Ausprägungen sind in Abschnitt 1.2 näher erläutert. Die Auswahl sollte von der Art des Systems, dem Nutzungskontext und der potentiellen Nutzer abhängig gemacht werden und kann nicht pauschal getroffen werden (vergleiche Oppermann 1994).

3 Einsatz

3.1 Beispiele

Adaptive Emailverwaltung

Ein Beispiel für die Übernahme von Routineaufgaben ist SwiftFile. Basierend auf der textuellen Analyse von Emails wird dem Benutzer eine vereinfachte Möglichkeit geboten, eine betrachtete Email in einen Ordner abzulegen. Dazu errechnet SwiftFile drei wahrscheinliche Zielordner und bietet diese dem Benutzer direkt zur Auswahl an, siehe Abbildung 3.

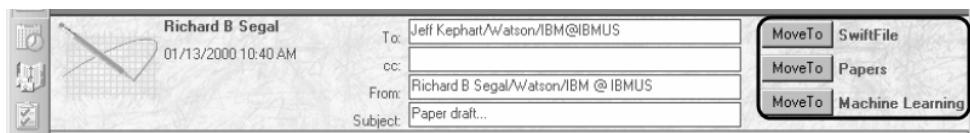


Abbildung 3: Adaptive Emailverwaltung, angepasst aus Segal & Kephart (2000)

Adaption der Displayhelligkeit und Beleuchtung

Ein Beispiel für die Adaption der Mensch-Maschine-Schnittstelle ist die Anpassung von Displayhelligkeit und Tastaturbeleuchtung abhängig von der Umgebungshelligkeit. Ein Sensor misst die Helligkeit der Umgebung, im System ist der Zusammenhang zwischen externer Helligkeit und einzustellender Displayhelligkeit und Tastaturbeleuchtung hinterlegt (Regelbasierte Adaption). Daher ist das System in der Lage seine Schnittstelle automatisch zu adaptieren.



Abbildung 4: Adaption der Displayhelligkeit und Beleuchtung (Bierkandt 2010)

In Abbildung 4 ist dies beispielhaft für ein mobiles Gerät dargestellt – bei heller Umgebung ist das Display relativ hell und die Tastaturbeleuchtung ist abgeschaltet, bei dunkler Umgebung ist das Display ebenfalls eher dunkel und die Tastaturbeleuchtung ist eingeschaltet.

Hilfestellung

Ein Beispiel für die Anzeige von Hilfestellungen bei der Nutzung eines Systems sind Assistenten, die das aktuelle Nutzerverhalten analysieren und passend dazu Hilfestellungen einblenden. Der Office-Assistent in Microsoft Office Word 2003 ist ein halb-automatisches Beispiel. Wie in Abbildung 5 gezeigt erfassst das System beispielsweise aktuell vom Benutzer eingegebenen Text und bietet darauf aufbauend Hilfestellungen an. Der Benutzer kann entscheiden, ob er die Hilfestellung annehmen möchte oder nicht.

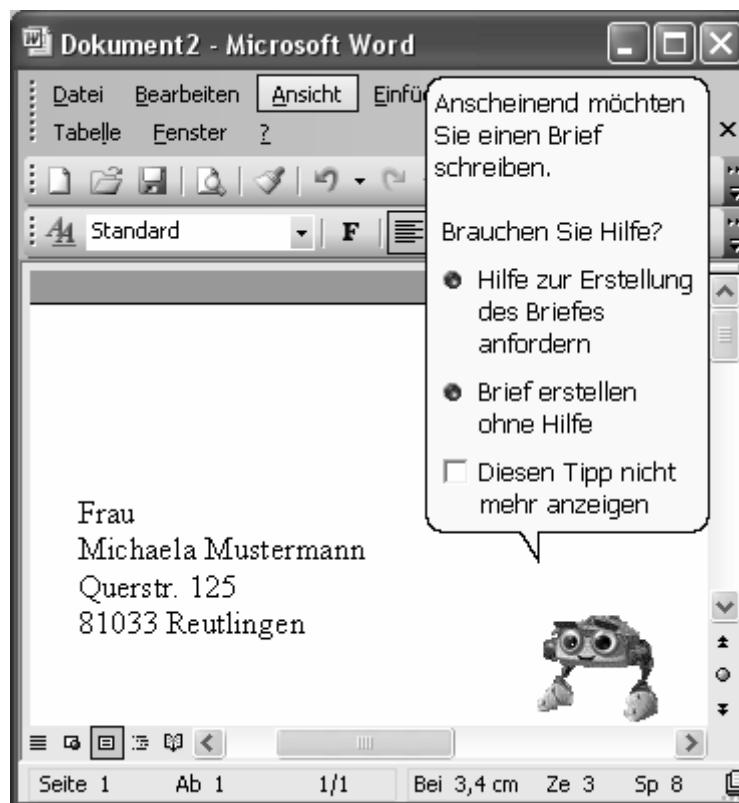


Abbildung 5: Hilfestellung zur Nutzung eines Systems (Bierkandt 2010)

4 Fazit

Die vorliegende Betrachtung von adaptiven Systemen zeigt, dass Adaptivität immer wichtiger wird (vergleiche Jameson 2003), da:

- vermehrt unterschiedliche Benutzer in immer unterschiedlicheren Situationen die Systeme nutzen wollen („Diversity of Users and Contexts of Use“)
- Benutzer meist nicht nur mit einem System arbeiten, sondern mit mehreren, die auch sehr komplex sein können. Daher können und/oder wollen sie ein System nicht vollständig verstehen („Number and Complexity of Interactive Systems“)
- Benutzer oftmals mit vielen Informationen und Objekten arbeiten müssen und die eigentlich triviale Aufgabe der Verwaltung dieser oft einen hohen Zeitaufwand birgt („Scope of Information to Be Dealt With“)

Jedoch haben adaptive Systeme hohes Potential sich zu entwickeln und zu verbessern. Zum einen ist die Gestaltung hinsichtlich Usability und einschlägiger Richtlinien eine große Herausforderung, wenn auch nicht unmöglich. Zum anderen beinhaltet die Interpretation des Verhaltens der Nutzer oft noch große Unschärfen. Hinzu kommt, dass Studien, deren Ziel die Analyse des Nutzens adaptiver Systemen ist, sich schwierig gestalten oder erst über längere Zeiträume hinweg belastbare und nutzbare Ergebnisse erbringen können, da sich die Benutzer erst an den Umgang mit adaptiven Systemen gewöhnen müssen und die Qualität der Adaptionen mit der Menge an gesammelten Daten steigt.

Wurde früher eher noch die Verwendung der klassischen human-centered Design Methode empfohlen (vergleiche Shneiderman & Maes 1997) und nicht die Verwendung adaptiver Systeme, so können sie heute für bestimmte, klar abgegrenzte Bereiche bereits eine gute Lösungsmöglichkeit sein. Dennoch sind weitere Forschungen und Untersuchungen notwendig, um Kosten und Nutzen von adaptiven Systemen und deren Usability weiter festzustellen und zu verbessern (Paymans et al. 2004).

Literaturverzeichnis

Bierkandt, J. (2009): Optimierung von Arbeitsabläufen durch neue HMI. Fraunhofer-Institut für Arbeitswirtschaft und Organisation: Human-Machine-Interfaces in der Industrie – Intuitive und innovative Gestaltung für effiziente Bedienung (Seminar). Stuttgart.

Bierkandt, J.; Bierkandt, M. (2010): Eigene Quelle

- Browne, D.; Totterdel, P.; Norman, M. (1990): Adaptive user interfaces. London: Academic Press.
- Carroll, J.; Rosson, M. B. (1989): The Paradox of the Active User. Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. Cambridge: MA, MIT Press.
- Dieterich, H.; Malinowski, U.; Kühme, T.; Schneider-Hufschmidt, M. (1993): State of the Art in Adaptive User Interfaces. In Adaptive User Interfaces, Kapitel 1, S. 13-48. Elsevier Science Publishers B. V.
- Gajos, K. Z.; Everitt, K.; Tan, D. S.; Czerwinski, M.; Weld, D. S. (2008): Predictability and Accuracy in Adaptive User Interfaces. Proceedings of the CHI 2008 Adaption. S.1271–1274. Florence, Italy.
- Hoppe, H.; Plötzner, R. (1991): Inductive knowledge acquisition for a UNIX coach. In Mental Models and Human-Computer Interaction 2.
- ISO 9241-110 (2008): Ergonomics of human-system interaction - Part 110: Dialogue principles
- ISO 9241-210 (2010): Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems
- Jameson, A. (2001): Systems that adapt to their Users. Tutorial presented at IJCAI 2001. Saarbrücken: German Research Center for Artificial Intelligence (DFKI).
- Jameson, A. (2003). Adaptive Interfaces and Agents. In Human-computer interaction handbook . S.305-330. Mahwah, NJ: Erlbaum.
- Kobsa, A. (1994): I User modeling and user-adapted interaction. Proceedings of CHI: Conference on Human Factors in Computing Systems. S.415-416. Boston: ACM.
- Kobsa, A.; Koenemann, J.; Pohl, W. (2001): Personalized Hypermedia Presentation Techniques for Improving Customer Relationships. In the Knowledge Engineering Review 16(2). S.111-155.
- Kobsa, A. (2004): Adaptive Methoden – Benutzermodellierung. Grundlagen der Information und Dokumentation (5th completely revised edition), S.299-302. München: K. G. Saur.

- Kühme, T. (1993): A user-centred Approach to adaptive Interfaces. In Intelligent User Interfaces. S.243-245.
- Künzer, A.; Ohmann, F.; Luczak, H. (2004): Anticipative User Modelling Based on Action Prediction Algorithms. Proceedings of the 7th International Conference on Work with Computing Systems, S.65-70
- Langley, P. (1997): Machine Learning for Adaptive User Interfaces. In Proceedings of the 21st German Annual Conference on AI.
- Leutner, D. (2002): Adaptivität und Adaptierbarkeit multimedialer Lehr- und Informationssysteme. In Information und Lernen mit Multimedia und Internet . S.115-125. Weinheim.
- Liang, T. (1987): User Interface Design for Decision Support Systems: A Self-Adaptive Approach. Information & Management Volume 12, Issue 4, April 1987, S. 181-193. University of Illinois at Urbana-Champaign: Elsevier Science Publishers B.V. (North-Holland)
- Mace, R. L. (1998): A perspective on universal design. In Designing for the 21st Century: International Conference on Universal Design.
- McGrenere, J., Baecker, R. M.; Booth, K. S. (2007): A field evaluation of an adaptable two-interface design for feature-rich software. In ACM Transactions on Computer-Human Interaction, Vol. 14, Nr. 1 Article 3: New York.
- Nitschke, J.; Wandke, H. (2001): Human Support as a Model for Assistive Technology. In: Heuer, A.; Kirste, T. (Ed.): Intelligent Interactive assistance and Mobile Multimedia Computing, S.130-136, Rostock: Neuer Hochschulschriftenverlag.
- Oppermann, R. (1994): Adaptively supported Adaptability. International Journal of Human-Computer Studies. Volume 4. S.544 – 472.
- Paymans, T. F.; Lindenbergs, J.; Neerincx, M. (2004): Usability Trade-offs for Adaptive User Interfaces: Ease of Use and Learnability. Proceedings of the IUI 2004. S.301-303. Madeira, Funchal, Portugal.
- Petko, D.; Reusser, K. (2006): Das Potenzial interaktiver Lernressourcen zur Förderung von Lernprozessen In E-Learning - Eine multiperspektivische Standortbestimmung (S.188) Bern: Haupt.

- Rauterberg, M. (1996): How to measure and to quantify usability of user interfaces. In Advances in Applied Ergonomics. S.429-432. West Lafayette.
- Segal, R. B.; Kephart, J. O. (2000): Incremental Learning in SwiftFile. Proceedings of the Seventh International Conference on Machine Learning.
- Shneiderman, B.; Maes , P. (1997): Direct manipulation vs. Interface.
- Shneiderman, B. (2003): Promoting Universal Usability with Multi-Layer Interface Design. Proceedings of the 2003 conference on Universal usability. Vancouver.
- Totterdell, P.; Rautenbach, E. (1990): Adaptation as a Problem of Design. In Browne, D.; Totterdell, P.; Norman, M.: Adaptive User Interfaces. S.61-84. Academic Press.
- Wahlster, W. (2003): Adaptive Layout of Dynamic Web Pages. Akademische Verlagsgesellschaft Aka GmbH.

Konzeption eines dezentralen sozialen Netzwerkes basierend auf offenen Webtechnologien

David Schlichtenberger

Medien- und Kommunikationsinformatik, Hochschule Reutlingen

Betreuer: Prof. Dr.-Ing. Peter Hertkorn
Prof. Dr.-Ing. Wolfgang Keller

Abstract:

Der soziale Faktor im Web hat sich etabliert. In den letzten Jahren ist eine starke Veränderung des Konsumentenverhaltens der Benutzer sozialer Netzwerke und der Unternehmenspolitik der Betreiber sozialer Netzwerke zu beobachten. Das soziale Web öffnet sich. Ziel der Master-Thesis war es, ein dezentrales soziales Netzwerk zu konzipieren. Dabei werden offene Webtechnologien, wie z.B. OpenID, OAuth, XRDS, Activity Streams und Mikroformate, eingesetzt. Ein Kommunikationsprotokoll wurde entwickelt, das Individuen plattformübergreifend verbinden kann. Zusätzlich wurden Aspekte der Konzeption in einem Prototyp realisiert. Das daraus resultierende Portal ermöglicht außerdem die Integration von bestehenden sozialen Webdiensten.

1 Einleitung

„The Web is now social and the Social Web is now open“ (Smarr 2009) verkündete Joseph Smarr in seiner Rede anlässlich der Google I/O Entwickler-Konferenz im Jahr 2009. Smarr deutete damit die Öffnung sozialer Netzwerke an, bis hin zu der Verwendung von offenen Webstandards. Ebenso wies er darauf hin, dass sich der soziale Faktor im Web etabliert hat.

Die Verwendung von sozialen Netzwerken hat sich für viele Nutzer im Alltag etabliert. Verschiedene Studien haben herausgefunden, dass viele Benutzer teils mehrere soziale Netzwerke verwenden (CSCM (2008), Seite 14). Zum einen liegt dies darin begründet, dass dadurch prinzipiell mehr Nutzer erreicht werden können. Zum anderen bieten verschiedene soziale Webdienste teils unterschiedliche Funktionen an und konzentrieren sich auf andere Schwerpunkte. Die Verwendung mehrerer sozialer Netzwerke kann für den Benutzer zu Problemen führen.

2 Probleme

Die Probleme, die sich bei der Verwendung mehrerer sozialer Netzwerke herausgestellt haben, sind u.a.:

- Repetition von Arbeitsabläufen

Darunter lässt sich verstehen, dass bestimmte Arbeitsabläufe, z.B. die Registrierung, das Verwalten von Einstellungen etc., für jeden sozialen Webdienst immer wieder neu vorgenommen werden müssen.

- Verwalten des sozialen Graphen

Der soziale Graph stellt die globale Vernetzung von Personen über Beziehungen dar. In der Praxis ist der soziale Graph auf mehrere Webdienste verteilt und somit fragmentiert (Fitzpatrick 2007). Für den Benutzer ergeben sich die Probleme, dass sie ihre Kontakte in jedem sozialen Netzwerk immer wieder neu hinzufügen und verwalten müssen. Sollte sich ein Benutzer dazu entschließen ein soziales Netzwerk nicht mehr zu verwenden, fallen im Ernstfall auch alle verbundenen Kontakte weg.

- Verteilung von Aktivitäten

Eine der wichtigsten Funktionen von sozialen Netzwerken ist der Austausch von Informationen und die Awareness über die Aktivitäten anderer (CSCM (2008), Seite 22). Die Mitgliedschaft in unterschiedlichen sozialen Netzwerken erschwert die Awareness, da es keinen globalen Aktivitätenstream gibt. Wenn ein Benutzer seine sozialen Netzwerke zu unterschiedlichen Zeiten pflegt, läuft er damit Gefahr, dass andere Benutzer davon nichts mitbekommen bzw. es erst zu spät erfahren.

Neben den vorgestellten Problemen, gibt es durchaus noch weitere Unzulänglichkeiten mit denen die Benutzer sozialer Netzwerke konfrontiert werden. In der Master-Thesis wird u.a. noch auf die Nutzung sozialer Webdienste auf mobilen Endgeräten und die Wiederverwendbarkeit von Daten eingegangen.

3 Anforderungen

Ziel der Konzeption des dezentralen sozialen Netzwerkes ist es, die oben genannten Probleme zu beseitigen. Eine Anforderung ist, dass das soziale Web dabei nicht neu erfunden werden soll, sondern dass die Konzeption auf bestehenden offenen Webtechnologien aufbaut und bisherige soziale Webdienste weiterhin verwendet werden können.

Dazu wurde zunächst überlegt in welche Bausteine man das soziale Web unterteilen kann. Diese sind:

- Identität
- Freunde/Kontakte
- Integration von sozialen Webdiensten

Ein Benutzer hat im sozialen Web genau eine Identität. Die Kontakte/Freunde des Benutzers sind direkt mit ihm verbunden. Über diese direkte Verbindung hat der Benutzer die Möglichkeit mit seinen Kontakten Aktivitäten auszutauschen. Aktivitäten sind definiert als alles was der Benutzer im sozialen Web macht (z.B. das Bereitstellen eines Fotos). Diese Aktivitäten stammen aus sozialen Webdiensten, die der Benutzer in seinem eigenen Portal integrieren kann.

Der dezentrale Ansatz besteht darin, dass jeder Benutzer ein Portal auf einem beliebigen Server installieren kann. Prinzipiell ist nicht vorgeben, wie das Portal aussieht bzw. wie es implementiert ist. Die Konzeption definiert lediglich ein Kommunikationsprotokoll zwischen den involvierten Parteien. Der Benutzer kann sich mit anderen Benutzern verbinden, die dasselbe Kommunikationsprotokoll verwenden und Aktivitäten austauschen. Des Weiteren kann er über sein Portal Informationen an verbundene sozialen Webdienste übermitteln bzw. aggregieren.

Das Kommunikationsprotokoll baut auf offenen Webtechnologien auf. Die Art wie diese zusammen verwendet werden, ist jedoch neu. Zum Einsatz kommen v.a. OpenID¹, OAuth², XRDS³, Activity Streams⁴, Mikroformate⁵ und PubSubHubbub⁶.

4 Umsetzung

Die Umsetzung der Konzeption erfolgte anhand eines Prototyps. Vorab wurde evaluiert, welche Technologien (Protokolle) sich für die einzelnen Bausteine einsetzen lassen und wie diese kombiniert werden können. Neben den Vorteilen, die sich für den Benutzer durch Lösung der oben dargestellten Probleme ergeben, ist es ebenso wichtig zu betrachten, welche Vorteile oder Geschäftsmodelle für die

¹ <http://openid.net>

² <http://oauth.net>

³ <http://www.oasis-open.org/>

⁴ <http://activitystrea.ms/>

⁵ <http://microformats.org/>

⁶ <http://code.google.com/p/pubsubhubbub/>

Betreiber der sozialen Webdienste möglich sind. Warum sollten diese sonst auch das konzipierte Kommunikationsprotokoll unterstützen? Daher wird in der Master-Thesis anhand von möglichen Beispielen erläutert wie Betreiber sozialer Webdienste von einem dezentralen Ansatz profitieren können.

Die Ergebnisse der Konzeption und der implementierte Prototyp haben gezeigt, dass ein dezentrales soziales Netzwerk auf Basis von offenen Webtechnologien umsetzbar ist.

5 Schluss

Das soziale Web kann immer nur eine Übersetzung sozialer Komponenten in technische Komponenten sein. Soziale Interaktion und Kommunikation bilden die Grundlage eines sozialen Systems. Daran lässt sich die Bedeutung von offenen Protokollen ableiten. Die menschliche Kommunikation folgt bestimmten Regeln, Verhalten und gemeinsamer Sprache, die zum Verständnis des Austauschs von Informationen führt. Ebenso werden soziale Webprotokolle benötigt, um Informationen auszutauschen (Schlichtenberger 2010).

Dezentralität ist das Grundprinzip des Internets und hat sich bereits in mehreren globalen Netzwerken etabliert. Der dezentrale Ansatz der Konzeption besteht darin, die Interaktion und Kommunikation zwischen sozialen Akteuren global im Internet zu ermöglichen. Die Grenzen sollen dabei nicht das jeweilige soziale Netzwerk einer einzigen Organisation darstellen.

Literaturverzeichnis

CSCM (2008): Erste Ergebnisse der Umfrage zur privaten Nutzung von Social Networking Services (SNS) in Deutschland.

http://www.kooperationssysteme.de/wp-content/uploads/republica-sns-umfrage_final_ppt03.ppt (07.09.2009).

Fitzpatrick, B. (2007): Brad's Thoughts on the Social Graph.

<http://bradfitz.com/social-graph-problem/> (07.09.2009).

Schlichtenberger, D. (2010): Konzeption eines dezentralen sozialen Netzwerkes basierend auf offenen Webtechnologien und Realisierung eines Prototyps. Master-Thesis. Reutlingen.

Smarr, J. (2009): Full video of my Google I/O talk now available.

<http://josephsmarr.com/2009/06/09/full-video-of-my-google-io-talk-now-available/> (05.09.2009).

Modell eines TOGAF basierten Portals für Enterprise Architecture Management (EAM)

Matthias Gutbrod, Felix Schiele, Christian Zinsenhofer

Wirtschaftsinformatik, Hochschule Reutlingen

Matthias.Gutbrod@Student.Reutlingen-University.de

Felix.Schiele@Student.Reutlingen-University.de

Christian.Zinsenhofer@Student.Reutlingen-University.de

Betreuer: Prof. Dr. rer. nat. Alfred Zimmermann

Abstract:

Das Konzept behandelt eine Metamodell gestützte Integration der Fachlichkeiten eines Enterprise Architecture Management Portals, welches mit Hilfe einer serviceorientierten Architektur umgesetzt wurde. Undurchsichtige, komplexe IT-Landschaften mit dezentraler Verwaltung stellen eine Herausforderung für die Unternehmen dar. Dieser kann durch die Einführung einer Enterprise Architecture begegnet werden. Bei immer kürzer werdenden Anpassungszyklen benötigen Unternehmen in ihrer Architektur einen hohen Grad an Agilität. Ein serviceorientiertes zentralisiertes EAM-Portal bietet die nötige Transparenz und Flexibilität. Einen solcher Ansatz bieten aktuelle Systeme wie PlanningIT, essential und iteratec noch nicht. In diesem Kontext stellt das beschriebene serviceorientierte EAM-Portal eine Innovation dar. Das EAM-Portal bietet dem Nutzer einen zentralen, personalisierbaren Einstiegspunkt in unternehmensweiten Architekturen.

1 Problemformulierung

Durch steigende Anforderungen sind in den Unternehmen komplexe IT-Landschaften mit dezentraler Verwaltung und mangelnder Transparenz entstanden. Redundante Systeme, Applikationen, Prozesse und steigender Verwaltungsaufwand führen zu hohen Betriebskosten für die IT. Diese gewachsene IT-Landschaft zeichnet sich durch starre Strukturen aus, was zu mangelnder Flexibilität führt. Laut M. Kütz (Kütz 2009) wird die IT in den Unternehmen oft als reiner Kostenfaktor gesehen, den es zu minimieren gilt. Änderungen der Business- und IT-Strategie erfolgen oft ohne ausreichende Planung und ohne Beachtung der Abhängigkeiten zwischen den Bereichen. Es existiert kein Wissen darüber, wie sich Änderungen auf der Fachseite und der IT auswirken. Wechselwirkungen und Konflikte können nicht frühzeitig identifiziert

werden, was eine optimale Abstimmung und Problemlösung verhindert. Für die zentrale Verwaltung der Unternehmensarchitektur bietet sich Enterprise Architecture Management an. Wichtig ist hierbei eine flexible Struktur, so dass die EAM Lösung schnell angepasst werden kann. Aktuell verfügbare EAM Lösungen von IDS Scheer, Alfabet, iteratec und EAS bieten diese Flexibilität mit Hilfe von serviceorientierten Integrationen noch nicht.

2 Einleitung

IT-Abteilungen sind häufig auf Konzeption, Entwicklung und Einsatz von Systemen, deren Architekturen und Technologien fokussiert. Neuere Architekturkonzepte erweitern diese zu einer Perspektive auf das Gesamtunternehmen und müssen daher Geschäftsarchitekturen, mit zugehörigen IT-Strategien und Betriebsarchitekturen mit einschließen. In diesem Kontext wird von einer Unternehmensarchitektur bzw. von einer Enterprise Architecture gesprochen (The Open Group 2009). Dieser erweiterte Blickwinkel hilft dabei Probleme zu lösen, welche mit einem eng gesetzten Fokus nicht sinnvoll zu lösen sind (Schwarzer 2009). Durch die ganzheitliche Sicht auf das Unternehmen kann die Lücke zwischen IT und Business bereits bei der Erstellung der Architektur verringert werden.

Die Enterprise Architecture setzt sich aus verschiedenen Teilarchitekturen zusammen. Im einfachsten Fall wird dabei die Geschäfts- und die IT-Architektur unterschieden. Die Geschäftsarchitektur umfasst dabei alle nicht-technischen Bausteine der Architektur. Die IT-Architektur umfasst hingegen alle technischen Bestandteile der Enterprise Architecture (Schwarzer 2009). Diese Zweiteilung ist in der Praxis nicht ausreichend, deshalb hat sich nach I. Hanschke (Hanschke 2009) eine Vierteilung der Unternehmensarchitektur als Best Practice erwiesen.

TOGAF (The Open Group 2009) stützt sich als wesentlicher international anerkannter Standart für EAM auf das TAFIM (Technical Architecture Framework for Information Management) des „Department of Defense“ (DoD) und wird seit 1995 ständig weiterentwickelt (Hanschke 2009). TOGAF ist ein umfangreiches Enterprise Architecture Framework, das verschiedene Methoden und Werkzeuge zur Entwicklung einer Enterprise Architecture zur Verfügung stellt. Dazu wird ein iteratives Prozessmodell, welches auf Basis von Best Practices erstellt wurde verwendet. Im Kern von TOGAF befinden sich vier Teilarchitekturen. Die Business Architektur definiert die Business Strategie, Steuerung, Organisation, und die Key Business Prozesse. Über die Data Architecture werden für das Unternehmen wichtigen Datentypen und Datenquellen definiert. In der Application Architecture werden wesentliche

Anwendungen definiert, welche für die Verarbeitung der Daten und die Unterstützung der Geschäftstätigkeit benötigt werden. Die Technology Architecture beschreibt die benötigten logischen Software- und Hardwareressourcen, die zur Unterstützung von Business, Daten und Applikationen benötigt werden. Wird eine Enterprise Architecture im Unternehmen eingeführt, muss sie fortlaufend gepflegt und weiterentwickelt werden. Dies geschieht unter Anwendung eines Enterprise Architecture Managements.

3 Das serviceorientierte EAM-Portal

Die Hauptidee des präsentierten Ansatzes ist das Konzept eines serviceorientierten EAM- Portals. Das EAM-Portal stellt den zentralen Einstiegspunkt für den Benutzer in das Enterprise Architecture Management dar. Ein Single Sign On ermöglicht den Zugriff auf verschiedene Applikationen. Das EAM-Portal bündelt durch eine konsistente, einheitliche und personalisierte Interaktionsmöglichkeit den ergonomischen Zugang für eine Vielfalt von angebotenen Funktionalitäten. Die Verwendung eines Rollenmanagements erlaubt die unternehmensweite Umsetzung von Sicherheitsaspekten.

Die vier Teilarchitekturen des TOGAF Frameworks umfassen die Business-, Daten-, Applikations- und Technologiearchitektur und bestimmen das Metamodell für die Modellierung der Enterprise Architecture. Ein Architecture Repository ermöglicht es alle wesentlichen durch das Metamodell definierten Artefakte konsistent abzulegen und diese Architekturbestände zu verwalten. Dadurch ermöglicht das EAM-Portal einen Gesamtüberblick über die ausbaubare Enterprise Architecture. Es bietet dem Benutzer eine zentralisierte Verwaltung aller Teilarchitekturen, wodurch die Transparenz im Unternehmen erhöht wird. Dies führt dazu, dass Redundanzen bei Geschäftsprozessen, Anwendungssystemen und den eingesetzten Technologien schneller identifiziert werden können. Darüber hinaus eröffnet die verwaltete Bestandsinformation der Enterprise Architecture die Simulation künftiger Ausbauszenarien der Informationssysteme bereits vor der Realisierung und zu einem frühen Zeitpunkt der Konzeption. Definierte und aufeinander ausgerichtete Teilarchitekturen der Enterprise Architecture ermöglichen ein konsequentes Business-IT-Alignment und damit konsistent und effizient abgestimmte Fachlichkeiten, Anwendungssysteme und IT-Infrastrukturen. Um die Unternehmensarchitekturen schnell an neue Anforderungen anpassen zu können, benötigt das EAM-Portal eine hohe Agilität. Eine serviceorientierte Realisierung des EAM-Portals bietet hierfür die notwendigen Eigenschaften zur effizienten Adaption der

Funktionalitäten sowie zur flexiblen Integration der Bestandsdaten. Die Implementierung des EAM-Portals soll durch Web Services mit der Java Enterprise Edition (Java EE) realisiert werden.

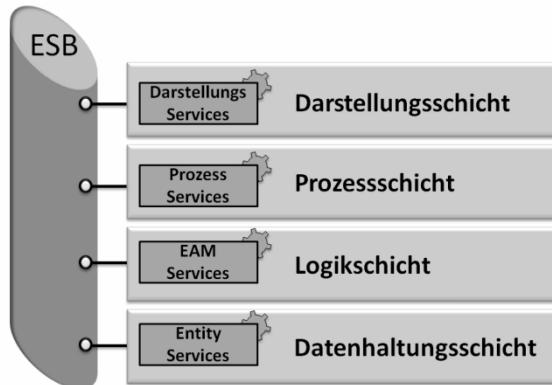


Abbildung 1: Serviceorientierte Architektur des EAM-Portals

In Abbildung 1 werden die Schichten des serviceorientierten EAM-Portals dargestellt. In der Darstellungsschicht befinden sich die Interaction-Services, welche die Interaktion mit dem Benutzer ermöglichen.

Für die Benutzungsoberfläche wird das Java Server Faces (JSF) Framework verwendet, welches das Model View Controller (MVC) Pattern implementiert (Starke & Tilkov 2007). Die Prozess-Services in der Prozessschicht übernehmen die Steuerung und Kontrolle der Prozesse des EAM-Portals. In diesen Prozessen werden die Task-Services der Logikschicht orchestriert. In der Logikschicht befinden sich die einzelnen Task-Services, welche die verschiedenen Grundfunktionen des EAM-Portals ermöglichen. Für den persistenten Datenzugriff werden spezifische Entity-Services eingesetzt. Dabei enthalten die Entity-Services neben den CRUD Funktionen auch entitätsspezifische Kernfunktionalitäten. Monitoring-Services unterstützen den Portalbenutzer durch Überwachungs- und Analysefunktionen von aktiven Services und liefern damit eine zusätzliche Transparenzfunktionalität über integrierte Systemlandschaften sowie konkret ausgeprägte Bearbeitungsfälle und Geschäftsprozesse. Utility-Services helfen in der Gesamtverarbeitung der Services durch allgemeine und mehrfach benötigte Basisfunktionen, beispielsweise für die Protokollierung von Ereignissen (Erl 2008) oder die Behandlung von fachlichen und technologischen Ausnahmen. Der Enterprise Service Bus (ESB) ist für die Kommunikation zwischen unterschiedlich konzipierten Services zuständig. Er koordiniert den Ablauf der Kommunikation und ermöglicht diese auch bei heterogenen Services.

und Landschaften indem er eine Kapselung der Nachrichten vornimmt (Starke & Tilkov 2007).

4 Schlussfolgerung und Ausblick

Im Rahmen dieses Projekts wurde ein serviceorientierter Entwurf eines Metamodell-gestützten EAM-Portals erstellt. Der Einsatz eines serviceorientierten EAM-Portals bietet dem Unternehmen die notwendige Flexibilität um die Unternehmensarchitektur effizient neuen Gegebenheiten anzupassen. Durch die Serviceorientierung verfügt das Portal über eine optimierte Anpassbarkeit, so lassen sich Änderungen mit geringem Aufwand in das EAM-Portal integrieren. Das Ergebnis der abgeschlossenen Konzeptions- und Entwurfsphase ist das beschriebene serviceorientierte Modell des EAM-Portals. Das Modell soll in weiteren Schritten durch Prototypen validiert und in einem Pilotprojekt erprobt werden. Gemäß durchgeföhrter Evaluierungen von alternativen Technologien und Infrastrukturen kann eine Realisierung auf Open Source Basis unter Verwendung des GlassFish Portfolios – der Referenzimplementierung von SUN für Java Technologien - erfolgen.

Literaturverzeichnis

Erl T. (2008): SOA: Entwurfsprinzipien für service-orientierte Architektur. München: Addison-Wesley Verlag, ISBN: 9783827326515

Hanschke, I. (2009): Strategisches Management der IT Landschaft. München: Hanser, ISBN: 9783446417021

Kütz, M. (2009): Kennzahlen in der IT. Heidelberg: dpunkt, ISBN: 9783898645799

Schwarzer, B.(2009): Einführung in das Enterprise Architecture Management. Norderstedt: Books on Demand GmbH, ISBN: 9783837021226

Starke, G.; Tilkov, S. (2007): SOA-Expertenwissen. Heidelberg: dpunkt, ISBN: 9783898644372

Stark, T. (2009): Java EE 5 - Einstieg für Anspruchsvolle. München: Pearson Studium, ISBN: 9783827326485

The Open Group (2009): TOGAF Version 9. Van Haren Publishing, ISBN: 9789087532307

Hoardingverfahren für eine Context-Aware Kalender Applikation

Roman Preuß, Rouven Alexander Rieker, Aleksandr Goncharov, Kateryna Shavrytska, Nazmi Tachiroglu, Dirk Marciniak

Wirtschaftsinformatik, Hochschule Reutlingen
Roman.Preuss@Student.Reutlingen-University.de
Rouven_alexander.Rieker@Student.Reutlingen-University.de
Aleksandr.Goncharov@Student.Reutlingen-University.de
Kateryna.Shavrytska@Student.Reutlingen-University.de
Dirk.Marciniak@Student.Reutlingen-University.de
Nazmi.Tachiroglu@Student.Reutlingen-University.de

Betreuer: Prof. Dr. rer nat Friedrich Laux

Abstract:

In Anwendungen für Smartphones - kurz: Apps - stellen einen rasant wachsenden Markt dar. Es gibt bereits eine Vielzahl von Apps, aber nur wenige nutzen das gesamte Potential mobiler Anwendungen. In dieser Ausarbeitung wird am Beispiel eines "Intelligenten Kalenders" eine kontextabhängige, mobile Anwendung vorgestellt. Sie sammelt automatisch Informationen und hält diese vor, um den Benutzer bei der Terminplanung besser zu unterstützen, als dies gewöhnliche Terminplaner vermögen. Der Kalender orientiert sich dabei an gewöhnlichen Bewegungsprofilen und dem aktuellen Standort, um das Terminmanagement zu optimieren.

1 Einleitung

Eine context-aware (kontextbewusste) Anwendung bezieht laut Roth (2005) Informationen über die Umgebung in die Verarbeitung mit ein. Als Kontext werden hier exemplarisch Termine und die Position eines Benutzers in einem mobilen, "intelligenten Terminplaner" verwendet. Dieser informiert den Benutzer zum einen schon beim Erfassen eines Termins, ob aufgrund des geographischen und zeitlichen Abstands der neue Termin überhaupt wahrgenommen werden kann. Zum anderen soll die Anwendung den Benutzer rechtzeitig, abhängig von seiner Position, an seine Termine erinnern.

Für die Implementierung wurde ein Smartphone mit dem Betriebssystem Android von Google ausgewählt, da es Open Source und somit frei verfügbar ist. Durch

den eingebauten GPS¹-Empfänger kann das Smartphone jederzeit seine Position und damit die Position des Benutzers ermitteln.

Die Ortsbestimmung hat zur Folge, dass die Kalenderanwendung ständig neue Verbindungsdaten² berechnen und abfragen muss. Dadurch entstehen einerseits hohe Ressourcenkosten³, welche die Akkulaufzeit des Endgerätes erheblich beeinträchtigen. Andererseits können auch höhere Verbindungskosten anfallen.

Diese Problematik kann durch den Einsatz von Hoardingverfahren⁴ und Caches verringert werden. Mit Hilfe eines Cache können erneute Anfragen auf dieselben Daten, lokal und somit kostengünstiger beantwortet werden (Höpfner et al. 2005, S.181). Wohingegen das Vorhalten von zukünftig benötigten Daten, was auch als Hoarding bezeichnet wird, dafür sorgt, dass diese Daten frühzeitig in den Cache geladen werden (Höpfner et al. 2005, S.207). In dieser Ausarbeitung wird ein Konzept für die Realisierung eines Hoardingverfahrens beschrieben und ein Anwendungsszenario vorgestellt.

2 Hoardingverfahren

Beim Horten von Daten (engl. data hoarding) lassen sich zukünftig benötigte Daten bereits im Voraus auf das Endgerät übertragen. So kann beispielsweise beim Laden der Daten darauf geachtet werden, dass ein günstiger WLAN-Zugang verwendet wird. Als Speicherort wird ein lokaler Cache benötigt. Alle Benutzeranfragen werden über diesen koordiniert, sodass der Benutzer nicht merkt, ob er im Offline- oder Onlinemodus arbeitet. Allerdings wirft diese Technik zwei Fragen auf:

1. Wie lassen sich zukünftig benötigte Daten eines Benutzers ermitteln?
2. Wie kann sichergestellt werden, dass diese Daten auch aktuell sind?

2.1 Daten ermitteln

Die Beantwortung der ersten Frage hängt von der Kenntnis über den zukünftigen Informationsbedarf ab. Dies kann manuell durch den Benutzer erfolgen oder automatisch ermittelt werden. Eine Reihe von automatischen Verfahren ist in Höpfner et al. (2005) beschrieben und in Kubach (2002) wird ein weiteres im Detail vorgestellt. Wir verwenden eine Variante des letzteren. Bei unserer Variante

¹ Global Positioning System, ein satellitengestütztes Orts- und Zeitbestimmungssystem

² Daten wie Abfahrtszeiten, Dauer, uvm., die zum Erreichen des Zielorts benötigt werden

³ In Form verbrauchter Rechenzeit und Stromverbrauch durch Einsatz von GSM, UMTS...

⁴ Vorhalten von zukünftig benötigten Daten ("horten")

werden Lokationsinformationen für die Bestimmung der zukünftigen Daten verwendet. Hierzu wird ein Bewegungsprofil des Benutzers dynamisch erstellt und aufbereitet (siehe 2.3 Bewegungsprofil). Mit diesen Daten, kann auf die zukünftigen Aufenthaltsorte des Benutzers geschlossen werden.

Im Zusammenhang mit dem "intelligenten Kalender" kann diese Eigenschaft so ausgenutzt werden, dass die Anwendung z.B. Fahrpläne vorab lädt, die zur Wahrnehmung eines Termins benötigt werden. Aufgrund dieser Möglichkeit kann der Nutzer zum einen, abhängig von seiner zukünftigen Position rechtzeitig an seinen Termin erinnert werden. Und zum anderen wird die ständige Neuermittlung der Positionsdaten überflüssig. Dies führt zu Ressourcen- und Kosteneinsparungen.

2.2 Aktualität der Daten

Die notwendige Aktualität der Daten hängt von der Art der Daten selbst und deren Anwendung ab. Werden die Daten für Datenbankänderungen benötigt, muss dafür gesorgt werden, dass die Datenbank ihren konsistenten Zustand beibehält (Tolia et al. 2007). Anders ist es hingegen, wenn die Daten nur lesend benötigt werden. In solchen Fällen ist es wichtig zu wissen, wann und wie häufig sich diese Daten ändern.

Als Beispiel sei hier eine Anwendung genannt, welche Börsenkurse eines Unternehmens bereitstellt. Sollen dem Nutzer die aktuellsten Kursinformationen angezeigt werden, ist es unmöglich, diese vorab zu übertragen. Soll dem Benutzer jedoch nur die Kursentwicklung der letzten Jahre gezeigt werden, können diese Daten vorab übertragen werden. Zumal sich diese Daten vermutlich nicht im Nachhinein ändern werden. Als weiteres Beispiel sei der Fahrplan von Linienbussen genannt. Diese Daten sind normalerweise über einen längeren Zeitraum gültig. Auch ist genau bekannt, wann sich diese Daten ändern werden. Daher kann die Aktualität der Daten bei unserem "intelligenten Kalender" in dem Sinne vernachlässigt werden, dass wir nur Verbindungsdaten von Bussen, Zügen und PKW benötigen, die sich zu bekannten Zeitpunkten ändern. Nur kurzfristige Fahrplanänderungen müssen gesondert behandelt werden.

Eine elegante Lösung des Problems wäre eine zusätzliche Server-Applikation, welche die Verbindungsdaten überwacht und die mobile Anwendung per Push-Dienst¹ über Änderungen informiert. Da der "intelligente Kalender" jedoch ohne einen zusätzlichen Server auskommen soll, kann dieses Verfahren nicht angewendet werden. Stattdessen sollen die Fahrpläne während der Online-Phasen

¹ Die Fähigkeit eines Endgerätes, sich über ein Ereignis informieren zu lassen

des Benutzers überprüft werden.

2.3 Bewegungsprofil

Mit einem Bewegungsprofil sind gesammelte Daten mit Orts- und Zeitbezug gemeint. Durch diese Daten lassen sich die Bewegungen des Benutzers nachvollziehen.

Erfasst werden diese Daten mit Hilfe eines GPS-Sensors, welcher in einer definierten Abtastrate¹ die aktuelle Position ermittelt. Diese Daten werden analysiert und zu Aufenthaltsorten zusammengefasst. So lässt sich einerseits die Datenmenge in Grenzen halten und zum anderen können mehrere Anfragen aus einem näheren Umkreis zusammengefasst werden. Beispielsweise ist es unnötig für jeden Punkt, welcher sich im Einzugsradius einer Bushaltestelle befindet, eine neue Anfrage zu starten.

Um mit dem Bewegungsprofil Aussagen treffen zu können, die das Horten von Daten ermöglichen, ist es notwendig, die gesammelten Daten aufzubereiten. Für das Aufbereiten der Daten eignen sich statistische Methoden. Für eine erste Implementierung wird der Modalwert² verwendet, dem ein Zeitraum zugeordnet wird. Hierzu werden alle gesammelten Daten verglichen und die Position, die am häufigsten vorkommt, wird als Modalwert für diesen Zeitraum genommen (Schwarze 2005). So lässt sich das Bewegungsverhalten des Benutzers in einem Tagesprofil festhalten und vorhersagen.

Weil sich ein Benutzer nicht jeden Tag gleich verhält, müssen mehrere Profile erstellt werden. Für unsere Anwendung muss mindestens ein Wochenprofil erstellt werden, in dem zwischen den einzelnen Wochentagen Unterschieden wird. Durch dieses Vorgehen, lässt sich die Qualität der Vorhersage steigern. Auch Profile, die Jahreszeiten und Feiertage berücksichtigen, wären ebenfalls denkbar und würden die Qualität, aber auch den Aufwand, steigern. Zur Vereinfachung wird hier angenommen, dass der Benutzer einen regelmäßigen Wochenrhythmus hat und sich daher jede Woche ähnlich bewegt.

In Abbildung 1 ist ein beispielhaftes Wochenbewegungsprofil eines Studenten zu sehen. Sie zeigt die Aufenthaltsorte, an welchen sich der Benutzer am häufigsten um diese Uhrzeiten aufhält. Der Bindestrich in manchen Feldern signalisiert, dass für diese Uhrzeit kein eindeutiger Aufenthaltsort bestimmt werden konnte. Dies ist dann der Fall, wenn der Benutzer zu einer bestimmten Uhrzeit viel unterwegs

¹ Bspw. jede Minute oder wenn der Benutzer sich um eine bestimmte Distanz bewegt

² Häufigster Wert

Uhrzeit\Tag	Mo	Di	Mi	Do	Fr	Sa	So
...
09:00:00	FH	FH	HOME	HOME	WORK	HOME	HOME
10:00:00	FH	FH	FH	-	WORK	HOME	HOME
11:00:00	FH	-	FH	HOME	WORK	HOME	HOME
12:00:00	FH	-	FH	FH	WORK	CITY	HOME
...

Abbildung 1: Grober beispielhafter Auszug eines (Wochen-) Bewegungsprofils

ist. Beispielsweise wenn ein Pendler über das Wochenende einen längeren Zeitraum nach Hause fährt. Allerdings handelt es sich in Abbildung 1 nur um eine grobe Übersicht, da hier eine Auflösung von einer Stunde gewählt wurde. Dies zeigt aber bereits auf, wie ein solches Bewegungsprofil funktioniert. Wird beispielsweise ein neuer Termin an einem Dienstag um 11:00 Uhr angelegt, berechnet die Anwendung die Reisezeit von der FH bis zum Zielort aus.

3 Ausblick

Um die Qualität der Vorhersage dieses Verfahrens zu verbessern, könnten Erfahrungswerte mit in die Berechnung einfließen. Mit Erfahrungswerten ist die tatsächlich benötigte Zeit für die vorgeschlagene Reiseverbindung zu verstehen. So könnten beispielsweise regelmäßige Verspätungen berücksichtigt und der Benutzer früher informiert werden. Auch könnten daraus Durchschnittswerte errechnet werden, mit welchen bereits im Offline-Modus, mit Hilfe der Luftlinienentfernung, eine erste Einschätzung der benötigten Zeit erfolgen könnte.

Literaturverzeichnis

Roth, J. (2005): Mobile Computing: Grundlagen, Technik, Konzepte. Heidelberg: dpunkt, 2., aktual. Aufl.

Höpfner, H.; König-Ries, B.; Türker, C. (2005): Mobile Datenbanken und Informationssysteme: Konzepte und Techniken. Heidelberg: dpunkt.

- Schwarze, J. (2005): Grundlagen der Statistik. NWB-Studienbücher Wirtschaftswissenschaften. Herne Verl. Neue Wirtschafts-Briefe, 10. Aufl. , S.69.
- Kubach, U. (2002): Vorabübertragung ortsbbezogener Informationen zur Unterstützung mobiler Systeme. Dissertation. Universität Stuttgart. Institut für Parallele und Verteilte Systeme. Fakultät Informatik, Elektrotechnik und Informationstechnik.
- Tolia, N.; Satyanarayanan, M.; Wolbach, A. (2007): Improving Mobile Database Access Over Wide-Area Networks Without Degrading Consistency, Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys'07), San Juan, Puerto Rico, USA.

Partner der Konferenz

Die Informatics Inside 2010 wird unterstützt von:

- Fraunhofer-Institut für Arbeitswirtschaft und Organisation
<http://www.hci.iao.fraunhofer.de>
- IBM Rational Software
<http://www.ibm.com/rational/>
- ict Innovative Communication Technologies AG
<http://www.ict.de>
- Max-Planck-Institut für biologische Kybernetik
<http://www.kyb.mpg.de/>
- photographerbook - Release your creativity
<http://www.photographerbook.eu/>
- Spectra Computersysteme
<http://www.spectra.de>/
- Stellwerk3 - Internet Regional
[http://www.stellwerk3.de/](http://www.stellwerk3.de)
- VDC Fellbach - Netzwerk für Virtual Engineering
<http://www.vdc-fellbach.de>

Wir bedanken uns bei den Partnern für die Unterstützung und freuen uns auf eine weiterhin gute Zusammenarbeit.