



Evaluation von Visualisierungskonzepten für Web-Animationen am Beispiel der "Process Change Patterns"

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Zied Khzami
zied.khzami@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Rüdiger Pryss

2012

„Evaluation von Visualisierungskonzepten für Web-Animationen am Beispiel der "Process Change Patterns"“
Fassung vom 20. Dezember 2012

DANKSAGUNGEN:

© 2012 Zied Khzami

Dieses Werk ist unter der Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License lizenziert: <http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

Satz: PDF- $\LaTeX$ 2_ε

Inhaltsverzeichnis

1	Einleitung	1
2	Zielsetzung	3
2.1	Ziel der Arbeit	3
2.2	Aufbau der Arbeit	4
2.3	Allgemeine Anforderungen	5
3	HTML5 vs. Flash	7
3.1	Vergleich zwischen HTML5 und Flash	8
3.1.1	Benutzersicht	9
3.1.2	Entwicklersicht	9
3.1.3	Plug-ins	9
3.1.4	Entwicklungsumgebung	10
3.1.5	Performanz	10
3.1.6	Digital Rights Management (DRM)	11
3.1.7	Grafiken und Fotos	11
3.1.8	Spieleprogrammierung	11
3.2	Adobe Flash	12
4	HTML5	13
4.1	Motivation	13
4.2	HTML5 Struktur	14
4.3	Aufbau des HTML5 Body	16
4.4	Überschriften in HTML5	18
4.5	Paragraphen in HTML5	18
4.6	Listen in HTML5	19
4.7	Tabellen in HTML5	20
4.8	HTML5 und Videos	20

Inhaltsverzeichnis

4.9	Canvas-Element in HTML5	21
4.10	Aktuelle Browser und deren Unterstützung von HTML5	23
5	CSS	25
5.1	CSS Syntax	26
5.1.1	Selektor	26
5.1.2	CSS Eigenschaften	27
5.1.3	Wertezuweisung	27
6	Flash Entwicklungsumgebung	29
6.1	Definition	29
6.2	Arbeitsfläche	29
6.3	Zusammenfassung	32
7	Change Patterns Anwendungsbeispiel	33
7.1	PAIS	33
7.2	Change Patterns	33
7.2.1	Allgemeine Design-Entscheidungen	34
7.2.2	Adaptation Patterns	35
	Hinzufügen und Löschen von Prozess-Fragmenten	35
	Verschieben und Ersetzen von Prozess-Fragmenten	37
	Hinzufügen und Löschen von Aktivitäten	38
	Anpassungen von Kontrollflusskanten	38
	Übergangsbedingungen Ändern	40
7.2.3	Änderungsmuster in vordefinierten Bereichen	41
8	Implementierung	45
8.1	Animationsübersichtsseite	45
8.2	Struktur der Animationsübersichtsseite	45
8.2.1	Header	45
8.2.2	Navigator	47
8.2.3	Sektion	47
8.2.4	Legende	47
8.2.5	Steuerkontrolle	47
8.2.6	Eventsyntax	48
8.2.7	Funktionssyntax	48

8.2.8	Code für die Button zur Animationskontrolle	48
8.2.9	Button „Description“	49
8.2.10	Textanimation	50
8.2.11	Hauptanimation	50
8.2.12	Animationen in HTML5 integrieren	51
8.3	Abgleich der gestellten Anforderungen	52
9	Zusammenfassung	55

1 Einleitung

Anfang der neunziger Jahre wurde Textverarbeitung durch leistungsschwache Geräte wie beispielsweise Schreibmaschinen realisiert. Aus diesem Grund waren Textformatierungen lediglich auf fette oder kursive Darstellungen beschränkt. Außerdem waren die Darstellungen nur auf dem Papier möglich. Im Laufe der Zeit wurden die Gestaltungsmöglichkeiten der Textformatierungen verbessert und die Darstellung von Tabellen, Bildern und Formeln konnten am Bildschirm vorbereitet und angezeigt werden.

Später wurden mit der Erfindung des WWW (World Wide Web) mehr Informationen (Dienste) in Form von Text, Bild, Audio, Video und Animation auf Servern abgelegt. Man greift auf diese Dienste mit Hilfe von Browsern zu. Die Webtexte, die sich auf Rechnern befinden und an das Internet angebunden sind, werden mit Hilfe von Style Sheets (CSS) in der Hyper Text Markup Language (HTML) formatiert.

Für die Darstellung von Video-Inhalten und Animationen im Browser werden Hilfsprogramme (Plug-ins) benötigt, die installiert werden müssen, um die Animationen laufen lassen zu können. Die bekanntesten Plug-ins sind Flash Player [1], Media-Player [2], Apples Quicktime-Plug-in [3] und Real Player [4]. Während man einige Plug-ins kostenlos im Internet herunterladen kann, muss man andere Vollversionen je nach Software käuflich erwerben. Die Plug-ins bieten die Möglichkeit, bewegte Bilder im Internet zu betrachten.

Das amerikanische Softwareunternehmen Macromedia war das erste Unternehmen, welches mit seiner führenden Authoring-Umgebung Flash, die Erstellung von Animationen unterstützt hat. Im Jahr 2005 wurde das Unternehmen Macromedia [5] durch das Unternehmen Adobe aufgekauft und dieses hat den Flash Player quasi als Internetstandard Plug-in für Animationen und Videos etabliert. Da im Laufe der Zeit die Internetverbindungen immer schneller wurden, entstanden zahlreiche Multimedia-Portale wie zum Beispiel Youtube [6] oder Dailymotion [7], die ihre Inhalte mit Hilfe des Plug-ins darstellen lassen konnten. Daher wurde Flash praktisch zum Webstandard für die Anzeige von Animationen und Videos. Um die Videos abspielen lassen zu können, müssen dafür alle Browser die benötigten Plug-ins

1 Einleitung

der Firma Adobe installieren. Weil sich mit Flash nicht nur Videos abspielen, sondern auch ganze Anwendungen programmieren lassen, wollte Apple Flash nicht unterstützen (und außerdem auch aus diversen anderen Gründen, die mit Flash zusammenhängen).

Apple [8] möchte das, was auf seinen Geräten läuft, möglichst weitgehend selbst steuern, um stabilere Anwendungen zu ermöglichen [44,45]. Dies geschieht, indem alle Programme für das iPhone und das iPad nur über den App Store erhältlich sind. Damit kann das Unternehmen nicht nur an jeder verkauften App Gewinn machen, sondern auch ihre Programme gegen Gefahren schützen, da das Flash-Plug-in Rechenpower sowie direkten Zugriff auf die Hardware braucht. Apple versucht hingegen, neue Web-Technologien wie HTML5 einzusetzen. Die neue Web-Technologie (HTML5) sagt dem „Flash-Plug-in“ den Kampf an.

2 Zielsetzung

2.1 Ziel der Arbeit

Ziel der Arbeit ist es, die Möglichkeiten von Flash mit HTML5 zu vergleichen. Dies geschieht an einem konkreten Beispiel - den Process Change Patterns [13]. HTML5 ist eine neue Technologie, bei der man für kleine Animationen einen großen Programmieraufwand betreiben muss. Leider findet man heutzutage noch keine guten Editoren für HTML5-Animationen. Bisher lassen sich am Softwaremarkt nur Editoren wie z.B "Sencha Animator" finden, die allerdings nur rudimentär in Beta Version vorliegen. Mit dessen Hilfe kann man ähnlich wie mit Flash, Animationen auf Basis von CSS3 für das Web realisieren. Die von Sencha Animator erzeugten Animationen werden von fast allen Geräten, auf denen ein WebKit [34] basierter Browser installiert ist, unterstützt. Dies gilt auch für das iPad und das iPhone. Sencha Animator stellt eine direkte Konkurrenz zu Adobe Flash dar. Leider ist Sencha Animator nur als Beta Version verfügbar, da er laut Webseite noch in den Kinderschuhen steckt.

In dieser Arbeit wird ein Vergleich zwischen HTML5-Canvas und Flash angestellt. Aus verschiedenen Gründen habe ich mich für Flash entschieden. Der wichtigste Grund jedoch war: HTML5-Canvas Editoren sind nur als Beta Versionen erhältlich.

Ich werde in der Arbeit am Anwendungsbeispiel der „Change Patterns“ [16,14] Flash umfassend für Animationen evaluieren. Dabei können die Animationen angehalten, pausiert oder erneut abgespielt werden (siehe dazu Abbildung 2.1). Anschließend werden die Animationen auf einer Webseite veröffentlicht.

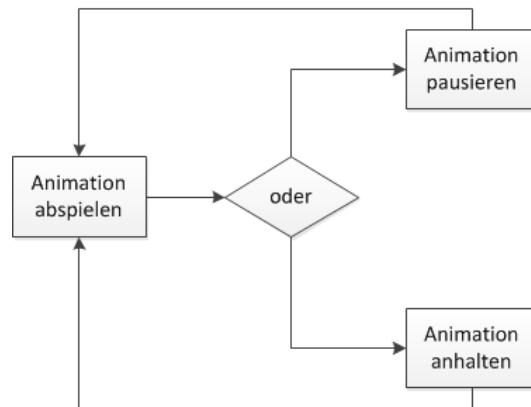


Abbildung 2.1: Animationssteuerung

2.2 Aufbau der Arbeit

In Kapitel 2 wird in das Thema Web-Animationen generell eingeführt und ein Überblick dazu gegeben.

In Kapitel 3 wird ein Vergleich der Technologien HTML5 und Flash angestellt. Dabei werden verschiedene Vor- und Nachteile von Flash und HTML5 diskutiert.

Kapitel 4 verschafft einen Überblick zu HTML5. Im selben Kapitel erfolgt die Darstellung einer Innovation von HTML5 gegenüber älteren HTML-Versionen. Außerdem werden die Struktur und die Syntax von HTML5 anhand zahlreicher Beispiele veranschaulicht.

Anschließend werden in Kapitel 5 Cascading Style Sheets (CSS) und die Syntax von CSS behandelt.

Das darauf folgende Kapitel 6 befasst sich mit den Grundlagen von Flash und bietet zusätzlich eine Übersicht über die Arbeitsfläche und die verschiedenen Formen von Tweens¹.

In Kapitel 7 werde ich auf die "Process Change Patterns" eingehen, welche im Verlauf der Arbeit animiert werden. Es werden dazu 18 konkrete Patterns, die von mir animiert wurden, zur besseren Veranschaulichung erläutert.

Abschließend werden einige Beispiele zu meiner Implementierung vorgestellt. In Abbildung 2.2 wird der gesamte Aufbau der Arbeit illustriert.

¹Beim klassischen Tween basiert die Animationszeitleiste auf Schlüsselbildern. Die Animation geschieht zwischen zwei Schlüsselbildern. Das Klassische Tween wird auf der Zeitleiste durch blau unterlegte Pfeile zwischen zwei Schlüsselbildern repräsentiert [34].

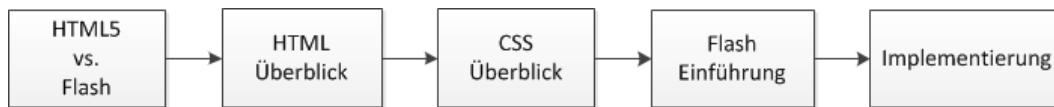


Abbildung 2.2: Aufbau der Arbeit

2.3 Allgemeine Anforderungen

Die allgemeinen Anforderungen für die Implementierung sind in der folgenden Tabelle zusammengefasst. Diese habe ich in der Arbeit entlang der Process Change Patterns erfüllt. Der Nutzer hat für die Animationen die Möglichkeit, die Animationen abzuspielen, zu pausieren oder erneut zu starten. Daher müssen wir die einzelne Komponenten (Texte, Aktivitäten, Kanten und Verknüpfungsoperationen (And-Split, And-Join, Or-Split, Or-Join)) der "Process Change Patterns" einzeln animieren. Schließlich sollen diese Animationen auf einer Webseite veröffentlicht werden (siehe Tabelle 2.1).

Allgemeine Anforderungen	
1. Animationen steuern	
	a. Animation abspielen
	b. Animation pausieren
	c. Animation stoppen
	d. Animation erneut abspielen
2. Animationen einzelner Prozessbestandteile	
	a. Animation von Texten
	b. Animation von Aktivitäten
	c. Animation von Kanten
	d. Animation von Operatoren (And-Split, And-Join, Or-Split, Or-Join)
3. Webseite erstellen	
	a. Animationen in eine HTML5-Webseite integrieren

Tabelle 2.1: Allgemeine Anforderungen

2 Zielsetzung

3 HTML5 vs. Flash

Auf jeder Webseite begegnet man Flash-Animationen, wie beispielsweise Werbungen und Nachrichten. Jedoch stellt sich die Frage, wie lange Flash-Animationen im Internet noch existieren werden und wie das Internet in zehn Jahren aussehen wird. HTML5 hat sich gegenüber Flash zum Hauptkonkurrenten entwickelt. Kann HTML5 wirklich Flash ersetzen? In diesem Abschnitt werde ich zunächst auf die Faktoren eingehen, die bis jetzt hilfreich waren für die Entwicklung von HTML5-Umgebungen. Anschließend werde ich einen Vergleich zwischen Flash und HTML5 ziehen.

Zurückblickend auf die letzten 5 Jahre haben sich die mobilen Endgeräte schnell verbreitet, darunter besonders Smartphones und Tablet-PCs, die überall aufs Internet zugreifen können. Diese Mobiltelefone haben besondere Eigenschaften im Vergleich zu den normalen Desktop-PCs. Sie bieten eine geringe CPU-Leistung und eine begrenzte Akku-Kapazität. Diese Eigenschaften erfordern schlanke Programme, die wenig Akkuleistung verbrauchen und geringe CPU-Leistung benötigen. Außerdem müssen die Programme, die auf dem Webbrowser laufen, schneller und unabhängiger von proprietären Programmen großer Softwareunternehmen sein. Deswegen basieren Smartphones auf Anwendungsprogrammen, den APPs. Das Wort App ist eine Abkürzung vom englischen Wort „Application“. Es bedeutet: Anwendungen, um den Funktionsumfang des Gerätes zu erweitern. Es gibt zwei Formen von Apps: Apps und Web-Apps. Mit Apps bezeichnet man Zusatzsoftware, die auf Smartphones installiert werden können (wie zum Beispiel Spiele). Mit Web-Apps oder auch Web-Applikationen bezeichnet man Anwendungen, die einen Browser zur Ausführung benötigen. Diese Anwendungen werden auf Basis von HTML, CSS und Javascript realisiert.

Web-APPs verarbeiten Webinhalte und verwenden Internetschnittstellen, um erzeugte Medieninhalte zu versenden. Es handelt sich also um Web-Applikationen, Webserver-Techniken und Webbrowser-Techniken, die zum Programmieren von Apps verwendet werden. Diese Techniken bieten die Möglichkeit, Plattformunabhängigkeit zu realisieren, d.h. die Anwendungen werden unabhängig von der Plattform, Hardware und vom Betriebssystem ausgeführt.

3 HTML5 vs. Flash

Viele Softwareunternehmen (z.B Apple) und Web-Entwickler kritisieren Adobe aus folgenden Gründen [35]:

- Flash Player Inhalte sind oft sehr langsam und erzeugen eine hohe CPU-Last.
- Flash stürzt häufig ab.
- Flash ist proprietär und verwendet keinen offenen Standard, obwohl die Webstandards für alle zugänglich sein sollten.
- Flash ist veraltet, da es nur alte Videoformate wie das SWF-Format unterstützt. Mittlerweile gibt es bereits neue Videoformate wie H.264 für mobile Geräte wie iPhones, iPods und iPads. H.264 ist eine neue Video Technologie, die im hohen Maße effiziente Videokompression unterstützt, d.h digitale Videos in höherer Qualität mit weniger Speicherplatzbedarf realisiert.
- Ebenso ist Flash mit vielen mobilen Geräten nicht kompatibel.
- Flash hat viele Sicherheitslücken, die von Angreifern missbraucht werden können, um beliebige Programmcodes auf fremden Systemen auszuführen.
- Bei Mobilgeräten verbraucht die Decodierung von Videos viel Akkuleistung. Beispielsweise kann die Akkulaufzeit von einem iPad 1,5 Stunden anstatt 10 Stunden bei einer Flash-Decodierung von Videos betragen.
- Flash unterstützt keine Touchscreen-Funktionen.

Diese oben aufgeführten Nachteile von Flash sind Gründe dafür, weshalb Apple sich gegen das Flash Plug-in wehrt und auch Flash nicht unterstützt.

3.1 Vergleich zwischen HTML5 und Flash

Wegen den oben genannten Mängeln von Flash wird im folgenden ein direkter Vergleich zwischen Flash und HTML5 anhand mehrerer Kriterien angestellt, um einen möglichen Umstieg auf HTML5 besser bewerten zu können.

3.1.1 Benutzersicht

Hinsichtlich der Video-Qualität ist für den Nutzer kein Unterschied ersichtlich (siehe Abbildung 3.1). Jedoch wurde bei HTML5 auf einen Plug-in Ansatz verzichtet, d.h man muss nicht mehr extra ein Plug-in installieren, um die Videos abspielen zu können.

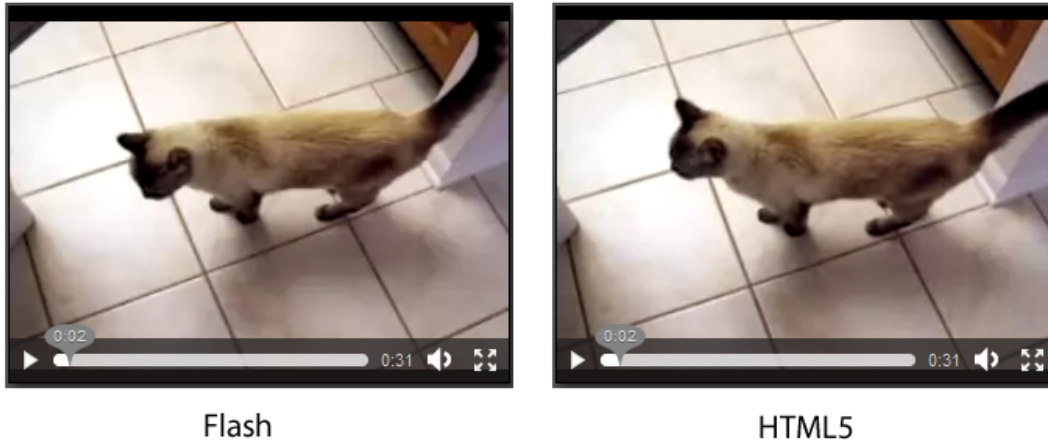


Abbildung 3.1: Video HTML5 vs Flash: vergleichbare Qualität

3.1.2 Entwicklersicht

Für den Nutzer ist die Videoqualität fast dieselbe, d.h es ist kein großer Unterschied zwischen HTML5 Videos und Flash Videos zu sehen. Im Gegensatz dazu ist für die Entwickler der Unterschied sehr groß. Flash ist proprietär, dagegen sind die Web-Standards für HTML5 offen. HTML5 ist Teil des Browsers, sodass eine HTML5 Animation auf verschiedenen Browsern robust und ohne zusätzliche externen Hilfsprogramme funktioniert.

3.1.3 Plug-ins

Plug-ins sind fremde Programme, die nicht integrierte Teile des Browsers sind, die aber durchaus nur teilweise unter Kontrolle des Browsers stehen. Dadurch können sie auch unabhängig vom Browser aktualisiert werden. Die Technik der Plug-ins wird auch von anderen Softwareunternehmen verwendet. Beispiele dafür sind das Adobe PDF Plug-in und Java Plug-ins. Plug-ins sollen die Aufgaben, die der Browser nicht leistet, realisieren: Der

3 HTML5 vs. Flash

Browser holt Daten aus dem Internet und übergibt sie dann an das Plug-in, welches die Daten bearbeitet und diese im Browser anzeigt. Im Gegensatz zu Plug-ins, ist HTML5 Browser-Funktionalität und steht unter seiner vollständigen Kontrolle. Daher bietet HTML5 mehr Flexibilität im Vergleich zu Flash. Abbildung 3.2 zeigt, dass ein Flash Plug-in installiert werden muss, um die Videos in Youtube anschauen zu können.

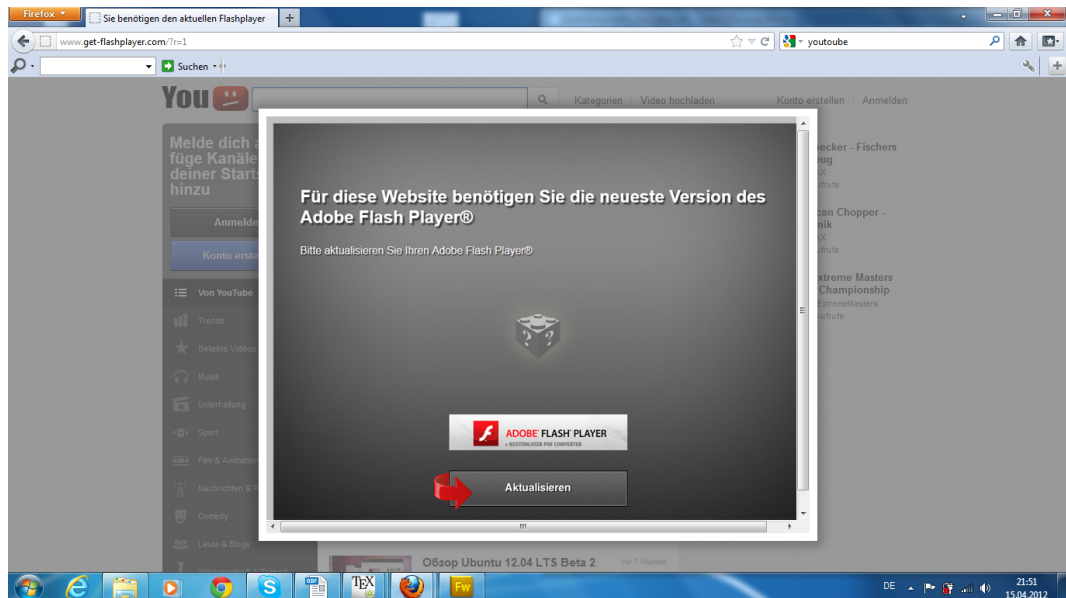


Abbildung 3.2: Flash Plug-in fehlt

3.1.4 Entwicklungsumgebung

Für HTML5- und Javascript-Entwicklungen benötigt man beliebige Texteditoren, die man kostenlos beziehen kann (wie Notepad, UltraEdit, TextPad, usw). Dagegen benötigt man für die Entwicklung von Flash die passende integrierte Entwicklungsumgebung (IDE), welche kostenpflichtig ist. Adobe Flash in einer Einzelplatzlizenz kostet beispielsweise ca. 700 Euro.

3.1.5 Performanz

Beim Abspielen von Videos mit Flash Player unter Linux oder Mac wird der Prozessor stark belastet, was zu Hardwarebelastung und schlechter Videoqualität führen kann. Dagegen ist

3.1 Vergleich zwischen HTML5 und Flash

die Qualität von Flash Videos unter Windows optimal. Zum Beispiel: für einen Mac Safari liegt die Auslastung der CPU mit Flash bei 10% bis 32 %, hingegen die Auslastung der CPU mit HTML5 nur bei 12%.

Bei Google Chrome liegt der CPU-Verbrauch für HTML5 als auch für Flash bei 50%.

Flash versucht die Geschwindigkeit dennoch zu optimieren, sodass auf dem Mac kaum noch Probleme auftreten.

3.1.6 Digital Rights Management (DRM)

Dies ist eine technische Maßnahme für die digitale Rechteverwaltung [36], mit der die Nutzung und die Verbreitung von digitalen Medien, vor allem von Filmen, Liedern und elektronischen Büchern, kontrolliert werden kann. Flash bietet für den Entwickler die Möglichkeit, seine digitalen Werke vor unerlaubter Verbreitung zu schützen. Der Entwickler kann ActionScript so anpassen, dass der Download des Videos nicht mehr zulässig ist [37]. HTML5 dagegen bietet bisher noch kein vergleichbares Verfahren, welches das Digital Rights Management ermöglicht.

3.1.7 Grafiken und Fotos

Bei der Darstellung von Bildern und Grafiken merkt man fast keinen Unterschied zwischen HTML5 und Flash. HTML5 bietet durch Canvas-Tags [38] in Kombination mit Javascript die Möglichkeit, komplexe Animationen zu erzeugen. Mit Javascript-Bibliotheken ist es auch einfach, beeindruckende Effekte im Browser zu realisieren.

3.1.8 Spieleprogrammierung

Spiele, die im Browser laufen, kann man genauso gut mit HTML5 wie mit Flash programmieren. Zwei wichtige Probleme bei der Programmierung von Online-Spielen sind zu erwähnen: Zum einen das unterschiedliche Verhalten von verschiedenen Webbrowsern und zum anderen die mangelhaften Tools für die Umsetzung. Flash hat dafür seinen eigenen Player entwickelt, um diese Probleme zu beseitigen. Dagegen ist HTML5 noch eine recht neue Technologie, weswegen von den Browser-Herstellern noch nicht alle Teile vollständig implementiert worden sind und man deshalb häufig noch auf Probleme stößt.

3.2 Adobe Flash

Am 9. November 2011 hat das Softwareunternehmen Adobe angekündigt, die Entwicklung des Flash Plug-in für mobile Geräte nicht mehr fortzuführen, und sich nur noch auf die Entwicklung von Flash Player für Desktop-PCs zu beschränken.

Adobe unterstützt auch HTML5 bei der Entwicklung von Web-Anwendungen. Es gibt zum Beispiel:

- Adobe Dreamweaver CS5 [9], ein HTML5-Editor, welcher von Adobe entwickelt wurde, um die Web-Entwicklung auf Basis von HTML5 zu realisieren.
- Adobe Muse [10], ein Werkzeug aus dem Hause Adobe zur Erstellung von Websites auf Basis von Adobe AIR [11]. Mit Adobe Muse werden automatisch Webseiten in HTML5, CSS und Javascript erstellt.
- Adobe Edge [39] ist auch ein Werkzeug aus dem Hause Adobe. Es ist ein Design Tool zum Erzeugen von animierten Webseiten, basierend auf HTML5, CSS3 und Javascript.

Trotz der rasanten Verbreitung von HTML5 Umgebungen in den letzten Jahren, bleibt Flash bis jetzt führend auf dem Softwaremarkt für Web-Animationen. Angesichts der Tatsache, dass auf den meisten Rechnern noch Flash Player installiert sind und auf den Servern Videos vorhanden sind, wird sich daran auch in den nächsten Jahren nicht viel verändern. Viele Videos, die auf den Servern liegen, sind auf Basis von Flash entwickelt worden und benötigen den Flash Player. Bis alle Videos aktualisiert sind und durch die neue Technik (HTML5) ersetzt werden, dauert es noch einige Zeit. Bis dahin müssen Flash Player für ältere Video Formate weiterverwendet werden.

Als Unterstützung von Adobe für die neue Technologie HTML5, hat das Softwareunternehmen die erste Testversion von Adobe Wallaby [10] veröffentlicht. Dieses Programm konvertiert Flash Dateien zu HTML5 Code, mit dem Ziel, eine Brücke zwischen der Adobe Flash Programmierung und HTML5 Programmierung zu bauen. Auf Apple Geräten laufen keine auf Flash basierenden Inhalte wie Filme und Animationen. Hier ist Adobe Wallaby sehr nützlich und wird eine große Rolle spielen, diese auf Flash basierenden Inhalte auf Apple Geräten laufen zu lassen. Aber leider befindet sich dieses Programm noch in einer frühen Entwicklungsphase.

4 HTML5

4.1 Motivation

HTML und CSS sind die Grundlagen für die Darstellung aller Internetwebseiten. Wenn man eine Webseite besucht, wird HTML5 in Zusammenarbeit mit CSS im Hintergrund interpretiert. Aus dem geschriebenen Quelltext (HTML- und CSS-Code) wird eine lesbare Internetseite generiert. Dies ist in Abbildung 4.1 illustriert.

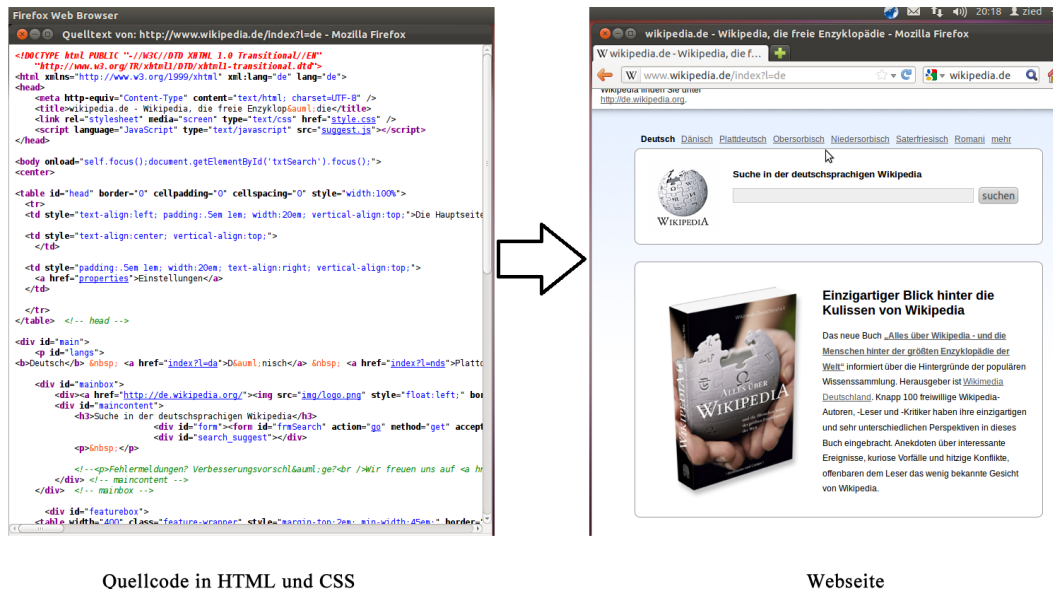


Abbildung 4.1: Aus dem geschriebenen Quelltext wird eine lesbare Internetseite generiert

HTML und CSS sind zwei komplementäre Sprachen, die miteinander harmonieren. Die Aufgabe von HTML ist die Verwaltung und Organisation von Webinhalten, das heißt, dass in HTML das geschrieben wird, was auf der Seite angezeigt werden soll: Texte, Tabellen, Bilder, Animationen, Links, usw. Der Web-Entwickler deutet den Titel, das Menü und wichtige Texte an. CSS hat die Aufgabe, das Aussehen der Webseite zu speichern (Layout,

4 HTML5

Positionierung der Bestandteile auf der Seite, Farbe, Schriftgröße, usw.; siehe dazu Abbildung 4.2.



Abbildung 4.2: Webseite ohne CSS

4.2 HTML5 Struktur

Ein HTML5 Dokument hat zwei wichtige Bereiche (siehe Abbildung 4.3):

- Zum einen den DOCTYPE-Bereich: Dies ist kein HTML-Tag, sondern eine Anweisung. Diese Anweisung teilt dem Internet-Browser mit, welcher Standard benutzt werden muss, welche HTML-Version verwendet wird, und welche Befehle der Browser erwarten kann. `<!DOCTYPE html>`.
- Zum anderen den HTML-Bereich: Dieser beginnt mit dem Tag `<HTML>` und endet mit dem Tag `</HTML>`. Er umfasst den Header-Bereich und den Body-Bereich:
 - `<Head>` Enthält die allgemeinen Informationen über die Seite. Wie zum Beispiel den Titel der Seite. Der Titel wird auf dem Browser-Kopf angezeigt. Im Head-Bereich stehen auch andere Metainformationen, die nicht auf der Webseite angezeigt werden, aber dennoch wichtig sind: Wie zum Beispiel die Codierung `<meta charset="utf-8" />`.

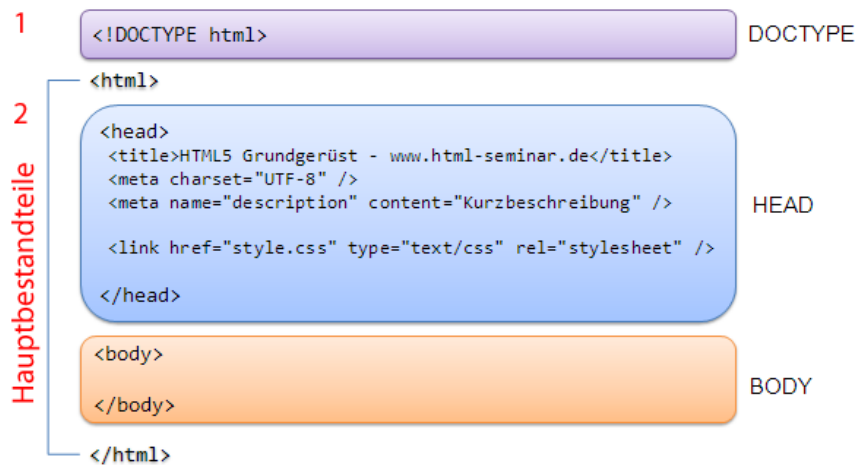


Abbildung 4.3: Aufbau einer HTML5 Seite

- `<Body>` ist der Hauptteil der Internet-Seite und enthält die Informationen, die auf der Seite angezeigt werden. Hier wird der größte Teil des Codes geschrieben. Alles was hier hinterlegt ist, wird auf der Seite angezeigt.

Beispielaufbau einer HTML5 Seite Listings 4.1 zeigt, wie eine HTML5 Seite generell durch den Header und den Body aufgebaut wird.

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <!-- der Header -->
5      <meta charset="utf-8" />
6      <title>Titel</title>
7    </head>
8
9    <body>
10     <!-- HTML Körper -->
11   </body>
12 </html>

```

Listings 4.1: Beispielaufbau einer HTML5 Seite

4.3 Aufbau des HTML5 Body

Der Body ist wie folgt strukturiert (siehe Abbildung 4.4):

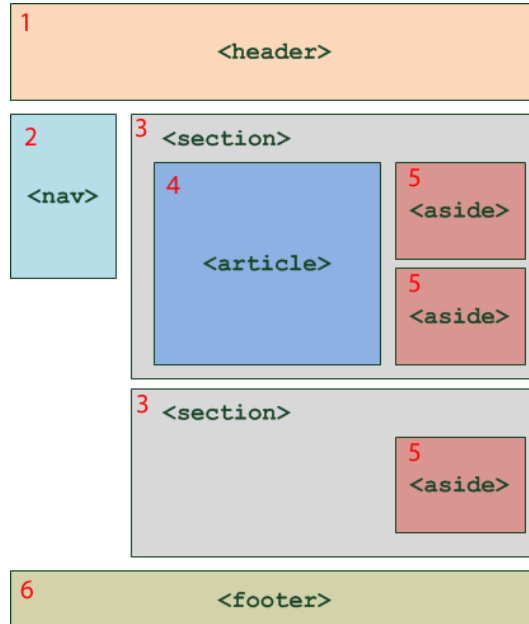


Abbildung 4.4: Aufbau des HTML5 Body

Konzepte von HTML5 sehen vor, die Webseiten mit einer besseren und semantischen Strukturierung zu ermöglichen. Das heißt, dass die Inhalte eine Bedeutung erhalten sollen. Es ist sehr nützlich, die Struktur einer Internetseite zu standardisieren, um die Blöcke, die durch das Element `<div>` definiert sind, übersichtlich zu halten. Zu diesem Zweck hat HTML5 neue Tags definiert: `<header>`, `<footer>`, `<nav>`, `<article>`, `<section>` und `<aside>`.

1. **`<header>`**: Die meisten Internetseiten haben in der Regel einen Kopfbereich. Dies ist ein Bereich, der allgemeine Informationen enthält: Titel der Webseite, Logos, Inhaltsverzeichnis, Formular zur Schnellsuche, usw. Alle diese Informationen werden im Tag `<header>` `</header>` definiert.
2. **`<nav>`**: Dient für die Hauptnavigation der Internetseite. Das Tag `<nav>` `</nav>` umfasst alle Hauptnavigationslinks der Webseite.
3. **`<section>`**: Ist dafür gedacht, den Inhalt der Internetseiten nach ihren Themen in unterschiedliche Bereiche zu gliedern. Jede Sektion kann einen Titel enthalten und

wird wiederum in die Bereiche Artikel `<article>` `</article>` und Aside `<aside>` `</aside>` unterteilt.

4. **<article>**: Das Tag `<article>` dient dazu, Inhalte des Tags unabhängig von den anderen Themen anzeigen zu lassen.
5. **<aside>**: Hat die Funktion, zusätzliche Informationen anzuzeigen. Diese zusätzlichen Informationen werden in der Regel an den Seiten platziert. Der Bereich `<aside>` ist optional und kann weggelassen werden.
6. **<footer>**: Fußzeile des Dokuments, die am unteren Rand des Dokuments steht und Informationen wie Links zu dem Namen des Autors enthält, um diesen kontaktieren zu können. Außerdem stehen hier rechtliche Hinweise. Diese Informationen stehen im Tag `<footer>` `</footer>` .

Beispiel der HTML5 Body Struktur

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Meine Webseite </title>
6    </head>
7    <body>
8      <header>
9        1 <h1>Title 1</h1>
10       <h2>Untertitel</h2>
11     </header>
12     <nav>
13       <ul>
14         2 <li><a href="#">Home</a></li>
15         <li><a href="#">Seite 1</a></li>
16         <li><a href="#">Seite 2</a></li>
17       </ul>
18     </nav>
19     <section>
20       <aside>
21         5 <h1>Author Informationen</h1>
22         <p> Herr X </p>
23       </aside>
24       3 <article>
25         4 <h1>Mein Urlaub in Brasilien</h1>
26         <p>Text steht hier</p>
27       </article>
28     </section>
29     <footer>
30       6 <p>Copyright...<br />
31       <a href="#">Meine Daten</a></p>
32     </footer>
33   </body>
34 </html>

```

Listings 4.2: Beispiel einer HTML5 Body Struktur

4.4 Überschriften in HTML5

HTML5 unterscheidet zwischen 6 Überschriftsebenen von 1 bis 6, um die Wichtigkeit des Titels auszudrücken (siehe Abbildung 4.5):

```
<h1>Überschrift vom Grad 1</h1>: Überschrift ist sehr wichtig.  
<h2>Überschrift vom Grad 2</h2>: Überschrift ist wichtig.  
<h3>Überschrift vom Grad 3</h3>: Überschrift weniger wichtig.  
<h4>Überschrift vom Grad 4</h4>: Überschrift noch weniger wichtig.  
<h5>Überschrift vom Grad 5</h5>: Überschrift nicht wichtig.  
<h6>Überschrift vom Grad 6</h6>: Überschrift nicht so wichtig, sieht wie  
ein normaler Text aus.
```

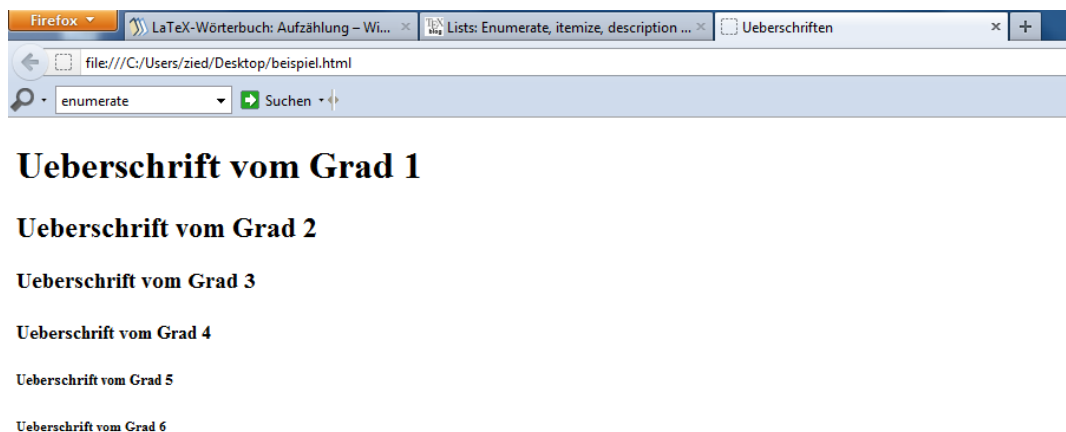


Abbildung 4.5: Überschriften in HTML5

4.5 Paragraphen in HTML5

Meistens werden die Texte in Absätze unterteilt. Die Absätze in HTML5 werden durch ein Tag `<P>` begrenzt. Das Zeichen `<P>` deutet den Beginn eines Paragraphen an und `</P>` bezeichnet auch das Ende eines Paragraphen (siehe Listings 4.3).


```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <title>Paragraph</title>
6  </head>
7
8  <body>
9      <p>Hier kann man sein Paragraph schreiben</p>
10 </body>
11 </html>

```

Listings 4.3: Paragraph

4.6 Listen in HTML5

HTML5 stellt dem Nutzer verschiedene Arten von Listen zur Verfügung:

- **Die Listenpunkte** werden durch ein Anfangs-Tag `<UL>` und ein Ende-Tag `</UL>` definiert und schließen einen oder mehrere Listenpunkte ein, die durch das Element `<li>` dargestellt werden. In Listings 4.4 werden die Listenpunkte einfach durch einen Punkt gekennzeichnet.

```

1  <ul>
2      <li>Listenpunkt 1</li>
3      <li>Listenpunkt 2</li>
4  </ul>

```

Listings 4.4: Listenpunkte

- **Die Zähllisten** werden durch das Anfangs-Tag `<OL>` und das Ende-Tag `</OL>` definiert und schließen einen oder mehrere Listenpunkte ein, die durch das Element `<li>` dargestellt werden. In Listings 4.5 werden die Listen entsprechend durchnummeriert.

```

1  <ol>
2      <li>Listenpunkt 1</li>
3      <li>Listenpunkt 1</li>
4  </ol>

```

Listings 4.5: Zähllisten

4.7 Tabellen in HTML5

In HTML5 sind Tabellen Zeile für Zeile aufgebaut und jede Zeile hat eine oder mehrere Spalten (siehe Listings 4.6).

Ein Tag `<table>` `</table>` gibt den Anfang und das Ende einer Tabelle an.

Ein Tag `<tr>` `</tr>` zeigt den Beginn und das Ende einer Tabellenzeile an.

Ein Tag `<td>` `</td>` zeigt den Beginn und das Ende einer Zeilenspalte an.

```
1 <table>
2   <tr>
3     <td>Zelle 1 in Zeile 1</td>
4     <td>Zelle 2 in Zeile 1</td>
5   </tr>
6   <tr>
7     <td>Zelle 1 in Zeile 2</td>
8     <td>Zelle 2 in Zeile 2</td>
9   </tr>
10 </table>
```

Listings 4.6: Tabellen in HTML5

4.8 HTML5 und Videos

HTML5 bietet die Möglichkeit, mithilfe des Tags `<Video>` , Videos in HTML-Seiten einzubinden und ohne Plug-in abspielen zu können (siehe Listings 4.7).

```
1 <video width="320" height="240" controls="controls">
2   <source src="movie.mp4" type="video/mp4" />
3   <source src="movie.ogv" type="video/ogg" />
4 </video>
```

Listings 4.7: Videos und HTML5

4.9 Canvas-Element in HTML5

Canvas ist auch eine Neuerung von HTML5. Das Tag `<canvas> </canvas>` dient dazu, Animationen mit Javascript zu erzeugen.

Beispiel: Code mit Javascript, um eine kleine Animation zu realisieren und ein Rechteck von links nach rechts zu bewegen. (siehe Abbildung 4.6).



Abbildung 4.6: Das Rechteck bewegt sich von links nach rechts

Listings 4.8, 4.9 und 4.10 zeigen den Quellcode für diese kleine Animation (ein Rechteck bewegt sich von links nach rechts) [41].

```

1  <!DOCTYPE HTML>
2  <html>
3    <head>
4      <style>
5        body {
6          margin: 0px;
7          padding: 0px;
8        }
9        #myCanvas {
10         border: 1px solid #9C9898;
11       }
12     </style>
13     <script>
14       window.requestAnimFrame = (function(callback) {
15         return window.requestAnimationFrame ||
16         window.webkitRequestAnimationFrame ||
17         window.mozRequestAnimationFrame ||
18         window.oRequestAnimationFrame ||
19         window.msRequestAnimationFrame ||
20         function(callback) {
21           window.setTimeout(callback, 1000 / 60);
22         };
23       })();
24
25       function animate(lastTime, myRectangle) {
26         var canvas = document.getElementById("myCanvas");
27         var context = canvas.getContext("2d");

```

Listings 4.8: Das Rechteck bewegt sich von links nach rechts

4 HTML5

```
29      // update
30      var date = new Date();
31      var time = date.getTime();
32      var timeDiff = time - lastTime;
33      var linearSpeed = 100;
34      // pixels / second
35      var linearDistEachFrame = linearSpeed * timeDiff / 1000;
36      var currentX = myRectangle.x;
37
38      if(currentX < canvas.width - myRectangle.width - myRectangle.borderWidth / 2) {
39          var newX = currentX + linearDistEachFrame;
40          myRectangle.x = newX;
41      }
42      lastTime = time;
43
44      // clear
45      context.clearRect(0, 0, canvas.width, canvas.height);
46
47      // draw
48      context.beginPath();
49      context.rect(myRectangle.x, myRectangle.y, myRectangle.width, myRectangle.height);
50
51      context.fillStyle = "#8ED6FF";
52      context.fill();
53      context.lineWidth = myRectangle.borderWidth;
54      context.strokeStyle = "black";
55      context.stroke();
```

Listings 4.9: Das Rechteck bewegt sich von links nach rechts (Fortsetzung)

```
57      // request new frame
58      requestAnimationFrame(function() {
59          animate(lastTime, myRectangle);
60      });
61  }
62
63  window.onload = function() {
64      var myRectangle = {
65          x: 0,
66          y: 50,
67          width: 100,
68          height: 50,
69          borderWidth: 5
70      };
71
72      var date = new Date();
73      var time = date.getTime();
74      animate(time, myRectangle);
75  };
76
77  </script>
78  </head>
79  <body>
80      <canvas id="myCanvas" width="578" height="200"></canvas>
81  </body>
82  </html>
```

Listings 4.10: Das Rechteck bewegt sich von links nach rechts (Fortsetzung)

4.10 Aktuelle Browser und deren Unterstützung von HTML5

Die meisten aktuellen Browser unterstützen HTML5 wie beispielsweise der Internet Explorer, Firefox, Opera, Safari und Chrome. Folgende Tabelle gibt eine genaue Übersicht über die gängigsten Browser, die HTML5 unterstützen.

HTML5 Kompatibilität													
	MAC						Windows						
	Safari	Firefox	Opera	Chrome		Safari	Internet Explorer			Firefox	Chrome		
Version	5,1	8	9	11,1	15	17	5,1	6	7	8	9	8	15
Canvas	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
Canvas Text	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
SVG	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
SVG Clipping Paths	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
SVG Inline	✓	✓	✓	x	✓	✓	✓	x	x	x	✓	✓	✓
SMIL	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
WebGL	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
Audio	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓
Video	✓	✓	✓	✓	✓	✓	✓	x	x	x	✓	✓	✓

Abbildung 4.7: HTML5 Kompatibilität in aktuellen Browsern [46]

5 CSS

CSS Code kann man an drei verschiedenen Stellen in einem HTML5 Dokument anbringen:

- Im Header (<Head>-Block) der HTML-Datei.
- Direkt in „Element-Tags“.
- In einer externen Datei mit der Endung .css. Die CSS Datei kann man in einer HTML-Datei durch ein Tag `<link rel="stylesheet" href="style.css" />` im Header referenzieren.

CSS Code wird meistens in einer externen Datei mit der Endung .css definiert. Diese Datei kann von jeder Webseite des Projekts aufgerufen werden. Mithilfe der externen Datei wird ein einheitlicher Stil für alle HTML-Dateien eines Projekts verwendet (siehe Abbildung 5.1). Diese Methode habe ich in meiner Arbeit zur Anzeige von Webseiten verwendet.

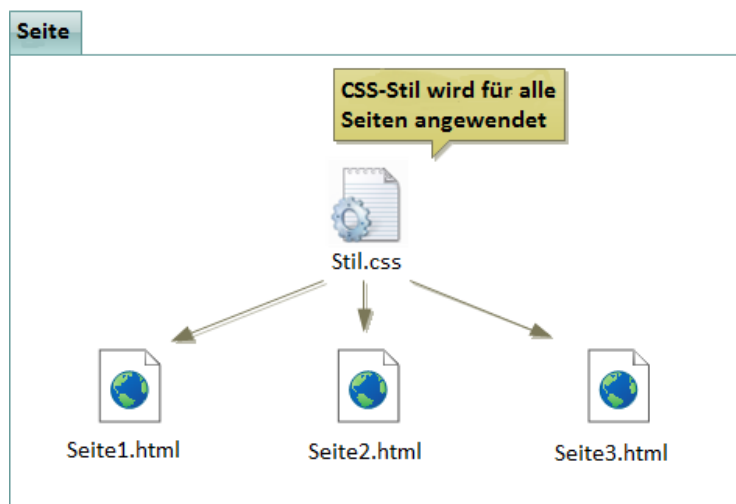


Abbildung 5.1: Aus dem geschriebenen Quelltext wird eine lesbare Internetseite generiert

5.1 CSS Syntax

Eine CSS Regel besteht aus drei wichtigen Elementen (Selektor, Eigenschaft und Wert) und ist wie im folgenden (siehe Listings 5.1) aufgebaut:

```

1  Selector [Selector1, Selector2,..., Selectorn]
2  {
3      Eigenschaft-1: Wert-1;
4      .
5      Eigenschaft-n: Wert-n;
6  }
```

Listings 5.1: SWF-Film in eine Webseite einbetten

5.1.1 Selektor

Ein Selektor ist immer vor dem Deklarationsblock definiert und legt fest, auf was der Deklarationsblock angewendet werden muss. In einem Selektor kann man verschiedene Auswahlkriterien verknüpfen:

- ***** : Wählt alle Elemente aus dem HTML-Dokument aus.
- **Elementname**: Wählt jedes Element mit dem gleichen Elementnamen aus dem HTML-Dokument aus.

Beispiel:

`P{ color : blue ; }` hier wird der definierte Stil auf alle Paragraphen des HTML-Dokumentes angewendet.

- **#ID**: Kennzeichnet ein bestimmtes Element im HTML-Dokument und wird im Quelltext als id-Attribut gesetzt. Dieses id-Attribut ist im ganzen Dokument einmalig, das heißt es darf nur einmal vorkommen.

Beispiel:

Im HTML-Dokument: `<p id="einleitung"> Die Einleitung ... </p>`

In der CSS Datei: `#einleitung{ color : blue ; }`

- **.klasse**: Damit kann man den definierten Stil auf mehrere Elemente (die einer Klasse angehören), aber nicht auf alle anwenden. Hierfür bietet sich die Nutzung von Klassen an. Eine Klasse kann im Gegenteil zu #ID in mehrere Elementen eines HTML-

Dokuments verwendet werden.

Beispiel: In CSS Datei wird eine Klasse wie folgt definiert: `.class_1{ color : blue ;}`

Im HTML-Dokument wird die bereits in CSS definierte Klasse wie folgt aufgerufen und auf den gewünschten Paragraphen angewandt (siehe Listings 5.2):

```
1 <p class="class_1"> Paragraph 1 </p>
2 <p class="class_1"> Paragraph 2 </p>
```

Listings 5.2: Nutzung von Klassen

5.1.2 CSS Eigenschaften

CSS bietet verschiedene Eigenschaften, um die Elemente einer Webseite genau im Anzeigefenster des Browsers zu positionieren und die Schriftformatierung festzulegen. Einige Beispiele für Eigenschaften sind: Hintergrundfarbe, Schrift-Größe, Schrift-Art, Schrift-Farbe, Rahmen-Größe, Elemente-Positionierung, usw.

5.1.3 Wertezuweisung

Jeder CSS Eigenschaft kann mindestens einer oder mehrere Werte zugeordnet werden, wie beispielsweise für die Schriftarten: Arial oder Times New Roman. Für die Schriftfarbe muss man den Namen der Farbe festlegen, usw (siehe dazu Listings 5.3).

Im Listings 5.3 wird die Schriftart festgelegt, die Werte "Arial" und "Serif" sind für die Eigenschaft "font-family" zugeordnet. Dann wird schlussendlich die Schriftgröße bestimmt.

```
1 header h2
2 {
3     font-family: Arial, serif; // hier wird die Schrift-Art festgelegt: Arial + Serif
4     font-size: 1.1em;         // hier wird die Schrift-Größe festgelegt
5     margin-top: 0px;          // Element legt den Außenabstand eines Elementes zu einem
6                               // anderen Element fest
7     font-weight: normal;      // hier wird festgelegt wie fett eine Schrift sein soll
8 }
```

Listings 5.3: Werte

6 Flash Entwicklungsumgebung

In der Implementierung wurde Flash verwendet, um die Change Patterns [13] zu animieren. Dieses Kapitel befasst sich mit den Grundlagen von Flash und bietet zusätzlich eine kleine Einleitung über die Arbeitsfläche und die verschiedenen Formen von Tweens, Bewegungstweens und Klassischen Tweens.

6.1 Definition

Die Anordnung von Bildern, die beim Ablauf den Eindruck eines bewegten Bildes erzeugen, bezeichnet man als Animation. Werden kleine Details der Einzelbilder hinsichtlich des Nachbarbildes verändert, so entsteht ein flüssiger Übergang einer Bewegung. Demzufolge werden bei heutigen Animationsprogrammen häufig die sogenannten Schlüsselbilder verwendet, die sich als Referenzbild für weiterführende Animation eignen. Durch diese Animationstechnik wird versucht, eine Webseite lebendiger darzustellen, da Animationen immer mehr Verbreitung im Internet finden.

6.2 Arbeitsfläche

Die Arbeitsfläche bei Adobe Flash ist in vier Teilbereiche unterteilt (siehe Abbildung 6.1):

- *Die Bühne* ist der Hauptarbeitsbereich, indem man Grafiken und Animationen bearbeiten kann. Beim Exportieren von Flash in das SWF-Format wird alles, was auf dem Bereich der Bühne steht (Grafiken, Bilder, Texte, Animationen), angezeigt.
- *Die Werkzeugpalette*: Enthält sämtliche Werkzeuge, die der Entwickler zum Bearbeiten von Inhalten in Flash benötigt. Das heißt Texte schreiben, Grafiken zeichnen, färben, radieren und Elemente auf der Bühne selektieren.

6 Flash Entwicklungsumgebung

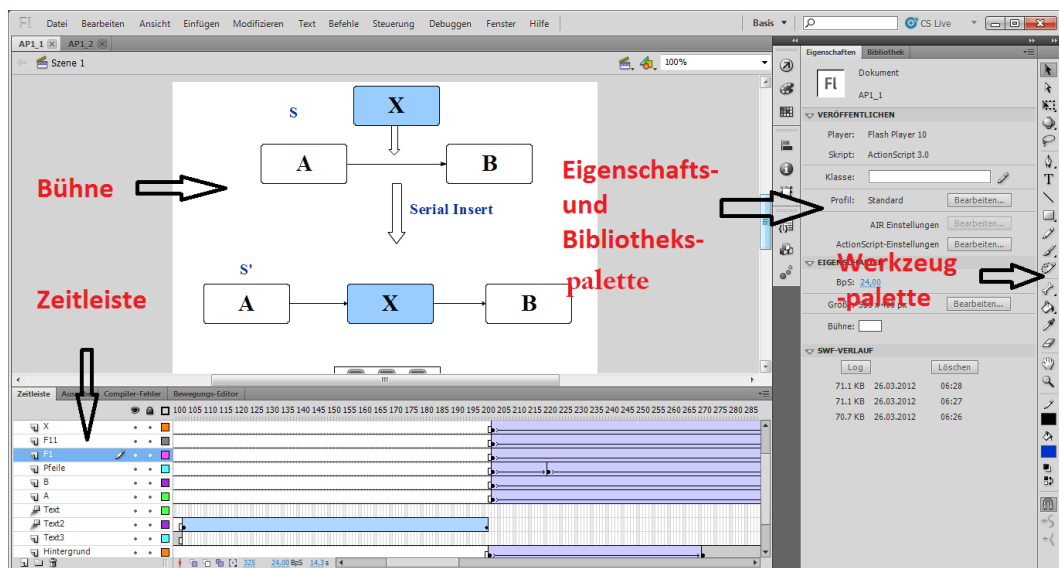


Abbildung 6.1: Arbeitsfläche

- *Die Zeitleiste:* Dient zur Steuerung und der Organisation des zeitlichen Ablaufs der Animation. Der Nutzer kann nach Bedarf die Zeitleiste in mehrere Ebenen unterteilen. Die Zeit wird über einzelne Bilder gesteuert. Man steuert dazu über die Zeitleiste und legt fest, wie viele Bilder pro Sekunde ablaufen sollen. Über die Zeitleiste lassen sich generell Elemente auf der Bühne animieren.
- *Die Eigenschaftspalette* dient dem Anzeigen von Elementeigenschaften und bietet die Möglichkeit, diese Eigenschaften zu verändern.
- *Bibliothek:* Jedes Objekt, welches auf der Bühne erstellt oder importiert wurde, wird in Objekte konvertiert und anschließend automatisch in der Bibliothek gespeichert. Das ist sehr nützlich, wenn man das gleiche Objekt mehrfach verwendet. Dazu muss man nur das gewünschte Objekt in der Bibliothek auswählen und per "Drag & Drop" auf die Bühne ziehen. Hiermit wird eine Instanz erzeugt. Jede Änderung auf dem Hauptobjekt bewirkt Änderungen auf alle erzeugten Instanzen. Auf diese Art kann man in nur einem Schritt viele Instanzen verändern. Wenn man außerdem viele Instanzen erzeugt, verändert sich die Dateigröße nicht. Das liegt daran, dass man die ganze Zeit nur mit einem Element arbeitet und es mehrfach verwendet.

Flash orientiert sich zum Erzeugen von Animationen an einer Reihe von Schlüsselbildern. Die Schlüsselbilder werden nacheinander abgespielt, sodass eine flüssige Bewegung ent-

steht. Wenn man die Animation Bild für Bild realisiert, ist es sehr mühsam, die einzelnen Bilder zu erstellen. Außerdem erhöhen die vielen Bilder den Speicherbedarf. Flash hat dieses Problem wie folgt gelöst: Es werden zwei Schlüsselbilder erzeugt, die den Anfangszustand und den Endzustand repräsentieren. Die dazwischenliegenden Bilder werden automatisch von Flash berechnet und auch erzeugt. Dieses Verfahren hat den Vorteil, einen geringen Speicherplatz zu erreichen.

Flash bietet zwei Arten von Animationen. Zum einen das "klassische Tween" und zum anderen das "Bewegungstween" : Beim klassischen Tween basiert die Animationszeitleiste auf Schlüsselbildern. Die Animation geschieht zwischen zwei Schlüsselbildern. Die Instanzen können entweder von demselben oder von verschiedenen Objekten sein. Das Klassische Tween wird auf der Zeitleiste durch blau unterlegte Pfeile zwischen zwei Schlüsselbildern repräsentiert (siehe Abbildung 6.2).



Abbildung 6.2: Klassisches Tween

Beim Bewegungstween basieren die Animationen auf einem Objekt. Die Animation basiert auf nur einem Zielobjekt für den gesamten Tweenbereich und nicht auf zwei Schlüsselbildern wie das klassische Tween. Bewegungstweens werden auf der Zeitleiste durch einen hellblauen Bereich repräsentiert (siehe Abbildung 6.3).



Abbildung 6.3: Bewegungstween

Für die beiden Arten, klassisches Tween und Bewegungstween, müssen Formen in Objekte konvertiert werden. Die Objekte werden in einer Bibliothek gespeichert (siehe dazu auch Abbildung 6.4).

6 Flash Entwicklungsumgebung

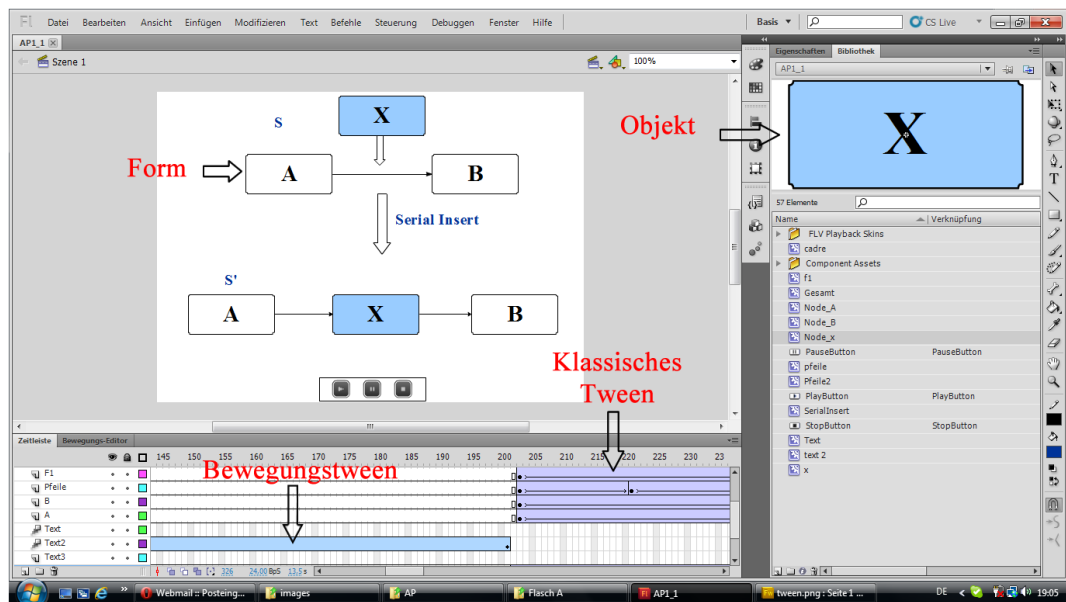


Abbildung 6.4: Objekte

6.3 Zusammenfassung

Flash bietet dem Nutzer eine bequeme Arbeitsfläche, um die Erstellung von Animationen zu erleichtern. Der Programmierer muss nicht Programmcode schreiben, er kann mit Objekten und Bildern direkt auf der Zeitleiste arbeiten. Mit dieser Vorgehensweise lässt sich viel Zeit sparen.

7 Change Patterns Anwendungsbeispiel

7.1 PAIS

Der Erfolg eines Unternehmen hängt davon ab, wie schnell und flexibel auf Änderungen reagiert werden kann [47,48]. Ursachen für diese Änderungen sind zum Beispiel die Einführung von neuen Gesetzen, Änderungen der Kundeneinstellungen, usw. Das Konzept eines PAIS (Process-Aware Information Systems) bietet vielversprechende Perspektiven, um diesen Erfolg zu realisieren [49]. PAIS verwaltet und führt operative Prozesse auf Basis von Prozessmodellen durch Einbezug von Personen, Anwendungen oder Informationsquellen aus und ist durch strikte Trennung von Prozesslogik und Anwendungscode charakterisiert. Beispiele für PAIS sind Workflow Management Systeme (WFM) [13] oder Business Process Management Systeme (BPM) [42]. Im Sinne eines Prozess-Modells beschreibt die Prozesslogik das Schema für die Prozessausführung. Prozess Management Systeme bieten generische Funktionalität für die Modellierung, Ausführung und Überwachung von Prozessen an. Die Trennbaren Blöcke in Prozessen sind eine gute Grundlage, um Änderungen isoliert durchzuführen. Dieses Verfahren ist sehr effizient zur Erhöhung der Wartungsfreundlichkeit und zur Reduzierung der Kosten im Fall von Änderungen.

7.2 Change Patterns

In diesem Kapitel werden 18 Änderungsmuster (Change Pattern) vorgestellt, die für die Änderungen des Kontrollflusses in PAIS relevant sind. Diese Muster behandeln verschiedene Lösungen für die am häufigsten auftretenden Veränderungen, welche durch PAIS realisiert werden müssen. Die Änderungsmuster werden in zwei Hauptkategorien unterteilt: Zum einen die Anpassungsmuster (**adaptation pattern**), die strukturelle Prozessanpassungen ermöglichen (insgesamt 14 Änderungsmustern). Zum anderen die Modelle für

Veränderungen in vordefinierten Regionen (**patterns for changes in predefined regions**), die integrierte Flexibilität ermöglichen (insgesamt 4 Änderungsmuster).

7.2.1 Allgemeine Design-Entscheidungen

Im Folgenden werden zwei allgemeine Design-Entscheidungen beschrieben, die für die Parametrisierung aller 14 Anpassungsmuster relevant sind [46]. Jedes Anpassungsmuster kann auf der Prozess-Typ und/oder Prozess-Instanz-Ebene (Auswahl A) angewendet werden. Darüber hinaus kann ein Anpassungsmuster auf einer atomaren Aktivität, einem Sub-Prozess oder in einer Hängematte (Auswahl B) durchgeführt werden (siehe Abbildung 7.1).

A. Was ist der Geltungsbereich der Muster?

- a) Prozess-Instanz-Ebene.
- b) Prozess-Typ Ebene.

B. Auf was kann man das Muster anwenden?

- a) Auf atomare Aktivitäten.
- b) Auf Sub-Prozesse.
- c) Auf einer Hängematte¹.

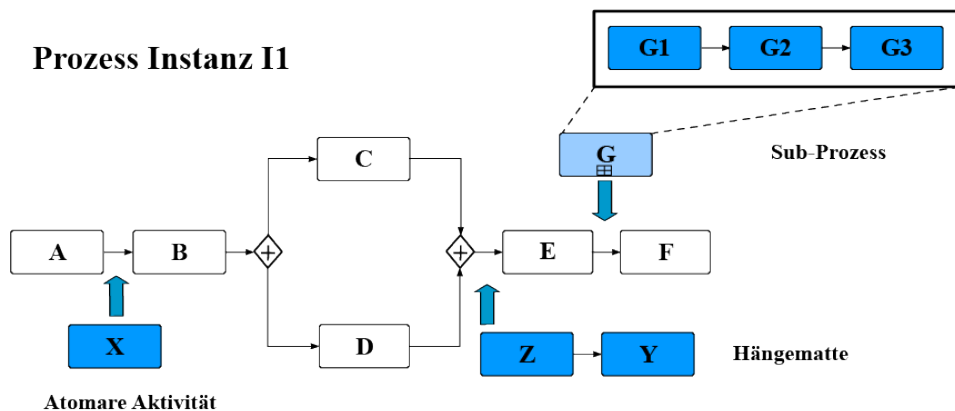


Abbildung 7.1: Design-Auswahl B

¹eine Hängematte ist eine Zusammenfassung von atomaren Aktivitäten.

7.2.2 Adaptation Patterns

Die Entwurfsmuster "Adaptation Pattern" dienen dazu, Prozess auf der Instanz-Ebene oder auf der Typ-Ebene strukturell zu verändern. Die strukturelle Veränderung von Prozessen geschieht auf Basis von „high-level change operations“ [50] (z.B eine Aktivität parallel zu einer anderen hinzufügen). Ziel ist Reduktion der Komplexität. Die Regionen, in denen die Adaptation Pattern angewendet werden, können dynamisch gewählt werden. Im allgemeinen Fall, wenn wir ein Adaptation Pattern auf ein Prozess-Schema S anwenden, erhalten wir dann ein neues Prozess-Schema S'.

Es existieren 14 Adaptation Pattern und diese sind wie folgt kategorisiert [15]:

- Hinzufügen und Löschen von Fragmenten in einem gegebenen Prozess-Schema wird durch AP1 (*Insert Process Fragment*) und AP2 (*Delete Process Fragment*) dargestellt.
- Verschieben und Ersetzen von Fragmenten wird durch AP3 (*Move Process Fragment*), AP4 (*Replace Process Fragment*), AP5 (*Swap Process Fragment*) und AP14 (*Copy Process Fragment*) dargestellt.
- Hinzufügen und Löschen von Fragmenten in einem gegebenen Prozess-Schema wird durch AP6 (*Extract Sub Process*) und AP7 (*Inline Sub Process*) dargestellt.
- Anpassung der Kontrolle wird durch AP8 (*Late Selection of Process Fragment*), AP9 (*Parallelize Activities*), AP10 (*Embed Process Fragment in Conditional Branch*), AP11 (*Add Control Dependency*) und AP12 (*Remove Control Dependency*) dargestellt.
- Ändern der Übergangsbedingung wird durch AP13 (*Update Condition*) dargestellt.

Hinzufügen und Löschen von Prozess-Fragmenten

AP1: Prozess-Fragment einfügen: Dieses Muster wird benötigt, um Prozess-Fragmente zu einem Prozess-Schema hinzuzufügen. Mit Hilfe von Optionen wird bestimmt, an welcher Position das neue Prozess-Fragment in dem jeweiligen Schema eingebettet wird. Bei manchen Systemen muss ein neues Prozess-Fragment zwischen zwei direkt aufeinanderfolgenden Aktivitäten hinzugefügt werden. Dieses wird "*Serial Insert*" genannt. Bei anderen Systemen muss man ein neues Prozess-Fragment zwischen zwei Gruppen von Aktivitäten unter Einhaltung bestimmter Randbedingungen hinzufügen. Dieses Verfahren

7 Change Patterns Anwendungsbeispiel

wird "Conditional Insert" genannt. "Parallel Insert" bildet einen Spezialfall zu "Conditional Insert" (siehe Abbildung 7.2).

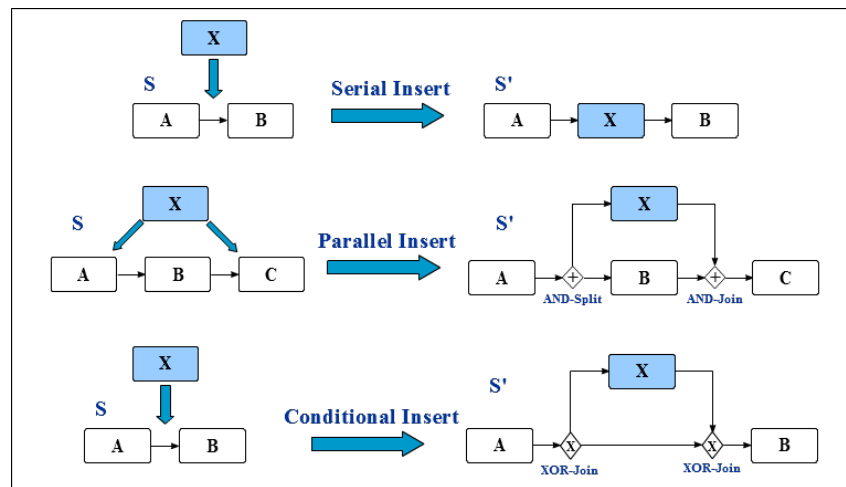


Abbildung 7.2: AP1: Prozess-Fragment einfügen

AP2: Prozess-Fragment löschen: Dieses Verfahren wird benötigt, um ein Prozess-Fragment von einem Prozess-Schema zu entfernen. Dafür gibt es drei Alternativen:

- Das betreffende Prozess-Fragment löschen. Das heißt: Die entsprechenden Knoten und Kanten löschen.
- Ersetzen des betreffenden Prozess-Fragment durch eine Aktivität, die keine Aktionen hat.
- Einbetten des betreffenden Prozess-Fragments in eine bedingte Verzweigung, die nie ausgeführt wird. Mit anderen Worten: Das Fragment bleibt Teil des Schemas, aber wird nie ausgeführt (siehe Abbildung 7.3).

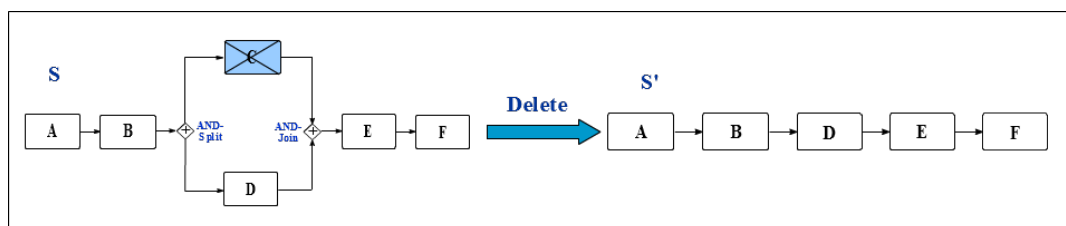


Abbildung 7.3: AP2: Prozess-Fragment löschen

Verschieben und Ersetzen von Prozess-Fragmenten

AP3: Prozess-Fragment bewegen: Anwendung dieses Musters gibt dem Benutzer die Möglichkeit, ein Prozess-Fragment von seiner aktuellen Position zu einer neuen zu verschieben. Eine zusätzliche Option bestimmt, wie das Fragment in dem Schema erneut eingebettet werden kann (siehe Abbildung 7.4).

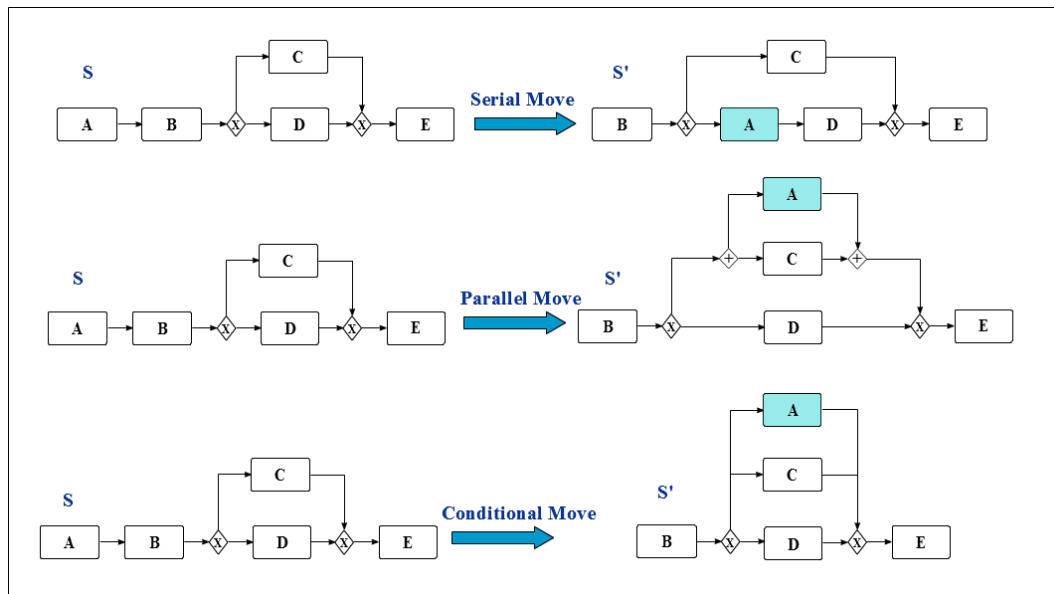


Abbildung 7.4: AP3: Prozess-Fragment bewegen

AP4: Prozess-Fragment ersetzen: Dieses Muster wird benötigt, um ein Prozess-Fragment durch ein anderes zu ersetzen (siehe Abbildung 7.5).

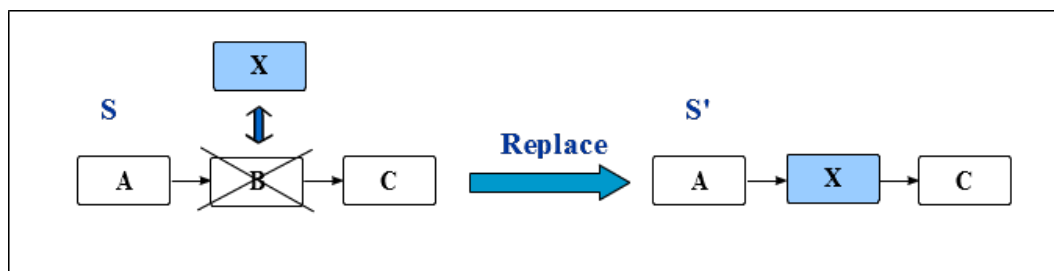


Abbildung 7.5: AP4: Prozess-Fragment ersetzen

7 Change Patterns Anwendungsbeispiel

AP5: Prozess-Fragment austauschen: Ermöglicht dem Benutzer, ein Prozess-Fragment durch ein anderes auszutauschen (siehe Abbildung 7.6).

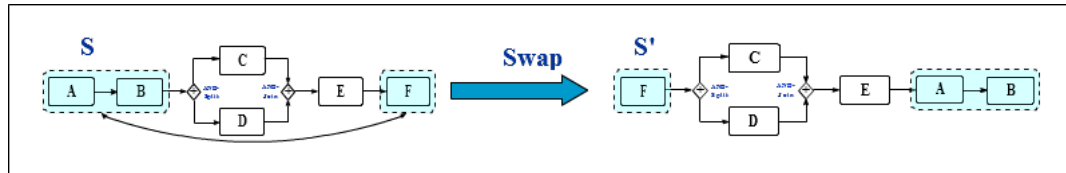


Abbildung 7.6: AP5: Prozess-Fragment austauschen

AP14: Prozess-Fragment kopieren: Dieses Muster ermöglicht dem Benutzer ein Prozess-Fragment zu kopieren (siehe Abbildung 7.7).

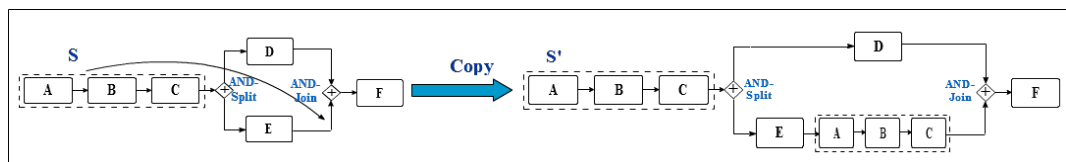


Abbildung 7.7: AP14: Prozess-Fragment kopieren

Hinzufügen und Löschen von Aktivitäten

AP6: Sub-Prozess extrahieren: Dieses Muster ermöglicht dem Benutzer, ein Prozess-Fragment aus einem Prozess-Schema zu extrahieren und anschließend das Prozess-Fragment in einem separaten Sub-Prozess-Schema zu kapseln (siehe Abbildung 7.8).

AP7: Inline Sub-Prozess: Ermöglicht dem Benutzer ein Sub-Prozess-Schema in den Eltern-Prozess zu integrieren. Dieses Verfahren verkleinert die Hierarchie eines Gesamtprozesses (siehe Abbildung 7.9).

Anpassungen von Kontrollflusskanten

AP8: Späte Auswahl eines Prozess-Fragments: Dieses Muster ermöglicht eine wiederholte Ausführung eines Prozess-Fragments. Die wiederholte Ausführung wird durch eine Schleife realisiert (siehe Abbildung 7.10).

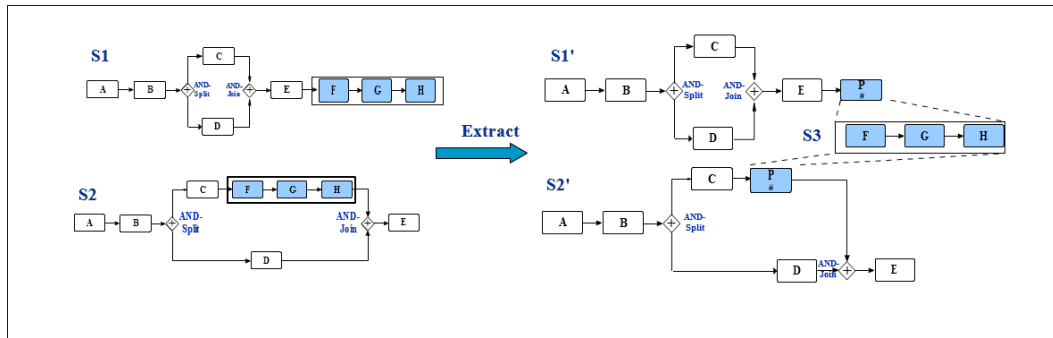


Abbildung 7.8: AP6: Sub-Prozess extrahieren

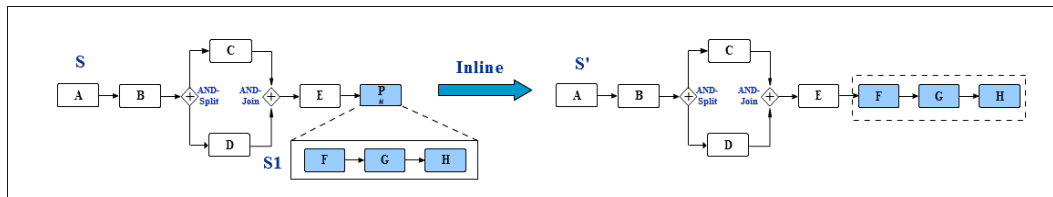


Abbildung 7.9: AP7: Inline Sub-Prozess

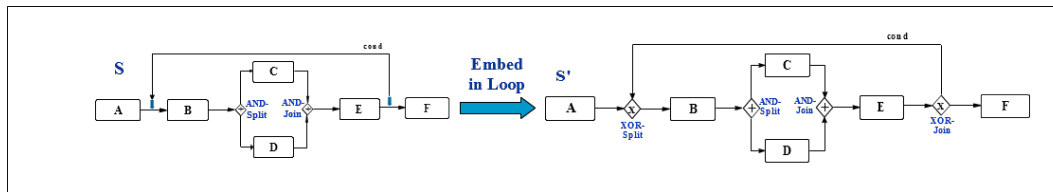


Abbildung 7.10: AP8: Späte Auswahl eines Prozess-Fragments

AP9: Aktivität parallelisieren: Ermöglicht die Parallelisierung Prozess-Fragmenten,

die beschränkt sind und nacheinander ausgeführt werden können (siehe Abbildung 7.11).

AP10: Einbetten eines Prozess-Fragments in bedingte Verzweigungen:

Mit diesem Muster kann ein vorhandenes Prozess-Fragment in einer bedingten Verzweigung ausgeführt werden, falls bestimmte Bedingungen erfüllt sind (siehe Abbildung 7.12).

AP11: Kontrollflusskante hinzufügen: Bei der Anwendung von AP11 wird eine Kontrollflusskante für ein gegebenes Prozess-Schema hinzugefügt (siehe Abbildung 7.13).

AP12: Kontrollflusskante löschen: Mit diesem Muster wird eine Kontrollflusskante aus einem Prozess-Schema entfernt (siehe Abbildung 7.14).

7 Change Patterns Anwendungsbeispiel

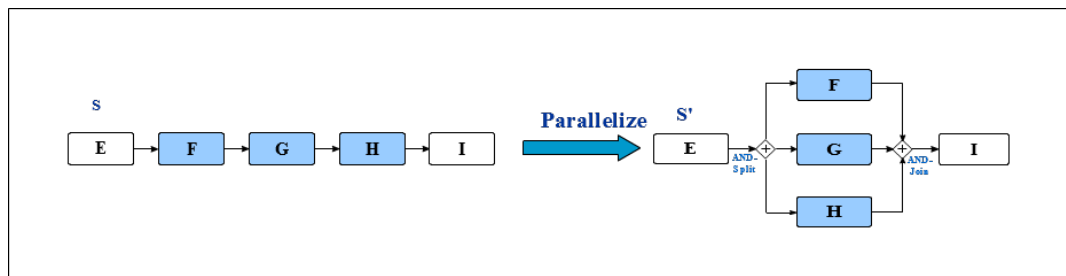


Abbildung 7.11: AP9: Aktivität parallelisieren

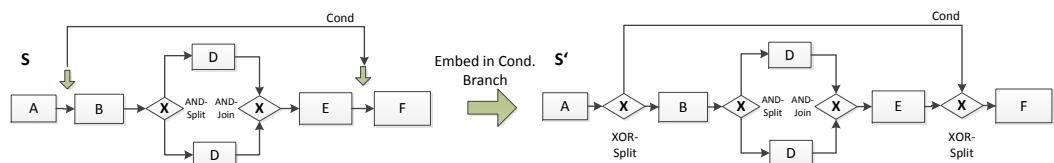


Abbildung 7.12: AP10: Einbetten eines Prozess-Fragments in eine bedingte Verzweigung

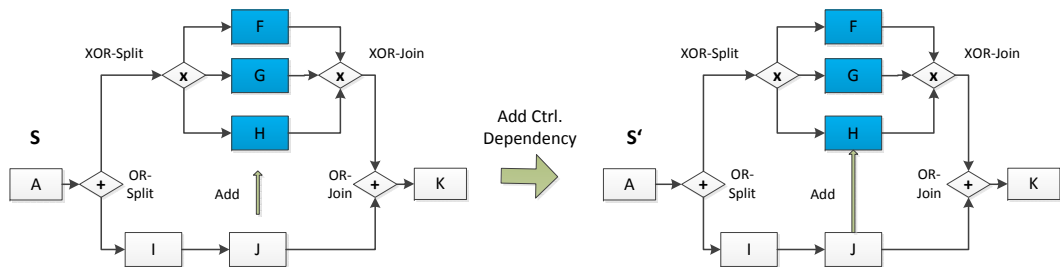


Abbildung 7.13: AP11: Kontrollflusskante hinzufügen

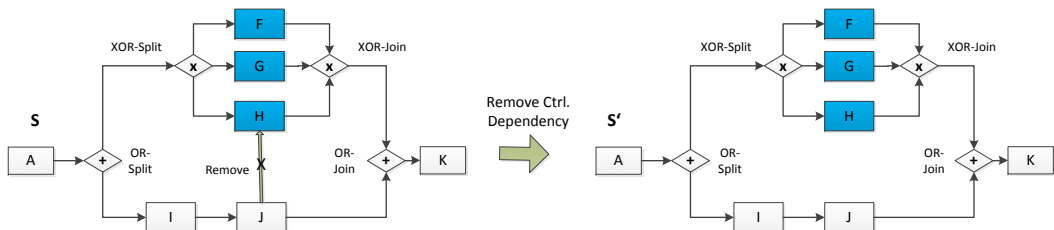


Abbildung 7.14: AP12: Kontrollflusskante löschen

Übergangsbedingungen Ändern

AP13: Zustand aktualisieren: Dieses Muster ermöglicht die Aktualisierung der Übergangsbedingungen einer Kontrollflusskante oder eines Gateways in einem Prozess-Schema (siehe Abbildung 7.15).

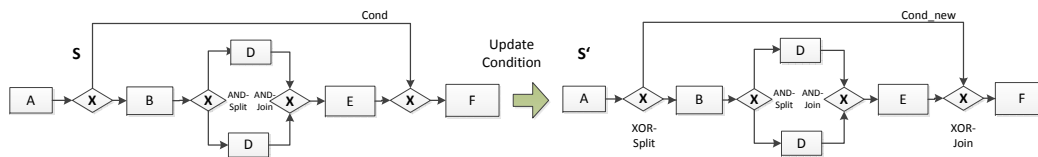


Abbildung 7.15: AP13: Zustand aktualisieren

7.2.3 Änderungsmuster in vordefinierten Bereichen

Mithilfe dieser Muster ist es möglich, Informationen in einem Prozess-Modell im Lauf der Zeit hinzufügen. Man definiert dazu Bereiche im Prozess-Schema. Nur in diesem vordefinierten Bereich können Änderungen oder Erweiterungen während der Laufzeit durchgeführt oder angewendet werden. In manchen Situationen kann man Entscheidungen über den genauen Ablauf zur Modellierzeit nicht treffen, und diese Entscheidungen müssen dann zur Laufzeit vorgenommen werden. Änderungsmuster in vordefinierten Bereichen sind gut geeignet für solche Situationen. Der Anwender lässt zuerst Teile des Modells un-spezifiziert, dann werden diese un-spezifizierte Teile später zur Laufzeit vervollständigt.

In dieser Kategorie lassen sich 4 Muster identifizieren: PP1 (*Late Selection*), PP2 (*Late Modeling*), PP3 (*späte Komposition von Prozess-Fragmenten*) und PP4 (*Multi-Instance Activity*).

PP1: Late Selection: Dieses Muster dient dazu, eine bestimmte Aktivität zur Umsetzung während der Ausführung (zur Laufzeit) auszuwählen. Dafür muss vor der Ausführung ein Platzhalter für die Aktivität zur Verfügung gestellt werden (siehe Abbildung 7.16).

PP2: Late Modeling: Dieses Muster ermöglicht die Modellierung von ausgewählten Teilen des Prozess-Schemas zur Laufzeit. Vor der Ausführung muss ein Platzhalter für die Aktivität zur Verfügung gestellt werden. Die Implementierung dieser Aktivität wird dann zur Laufzeit durchgeführt (siehe Abbildung 7.17).

PP3: Späte Komposition von Prozess-Fragmenten: Dieses Muster ermöglicht die Zusammensetzung von Prozess-Fragmenten aus dem Prozess-Repository (siehe Abbildung 7.18).

PP4: Multi-Instance Activity: Dieses Muster ermöglicht eine Verschiebung der Entscheidung, je nachdem, wie oft eine bestimmte Tätigkeit während der Laufzeit ausge-

7 Change Patterns Anwendungsbeispiel

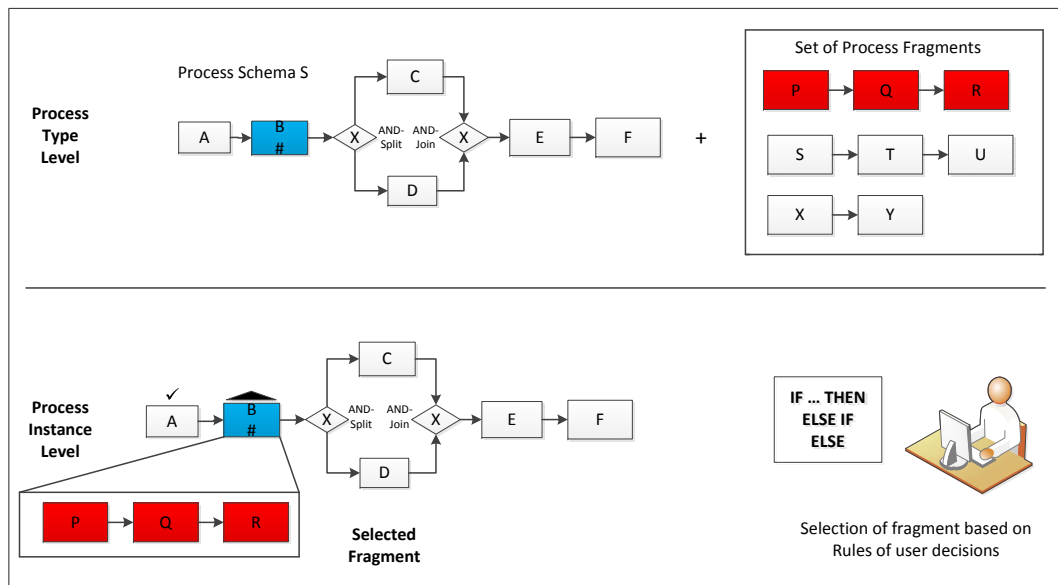


Abbildung 7.16: PP1: Late Selection

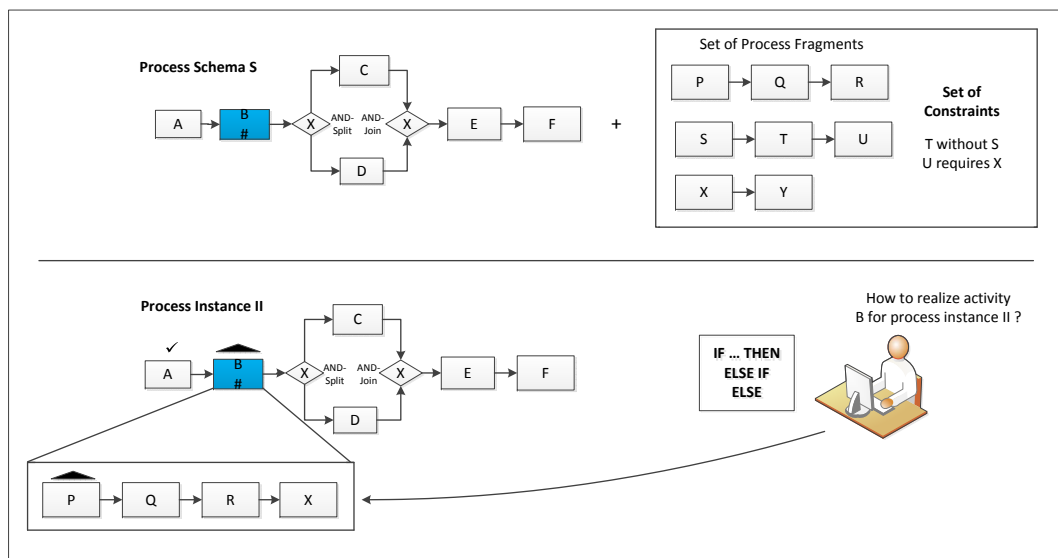


Abbildung 7.17: PP2: Late Modeling

führt werden soll. PP4 ermöglicht die Erstellung mehrerer Instanzen Aktivitäten zur Laufzeit (siehe Abbildung 7.19).

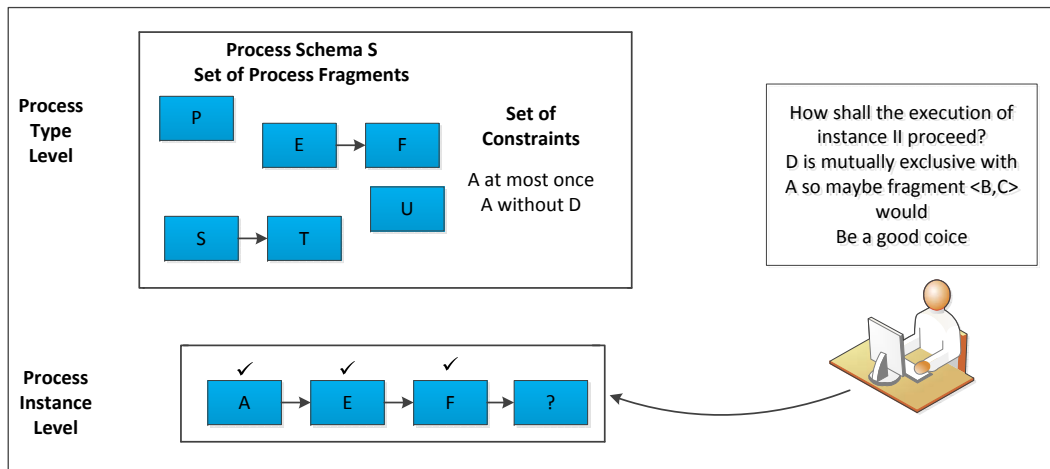


Abbildung 7.18: PP3: Späte Komposition von Prozess-Fragmenten

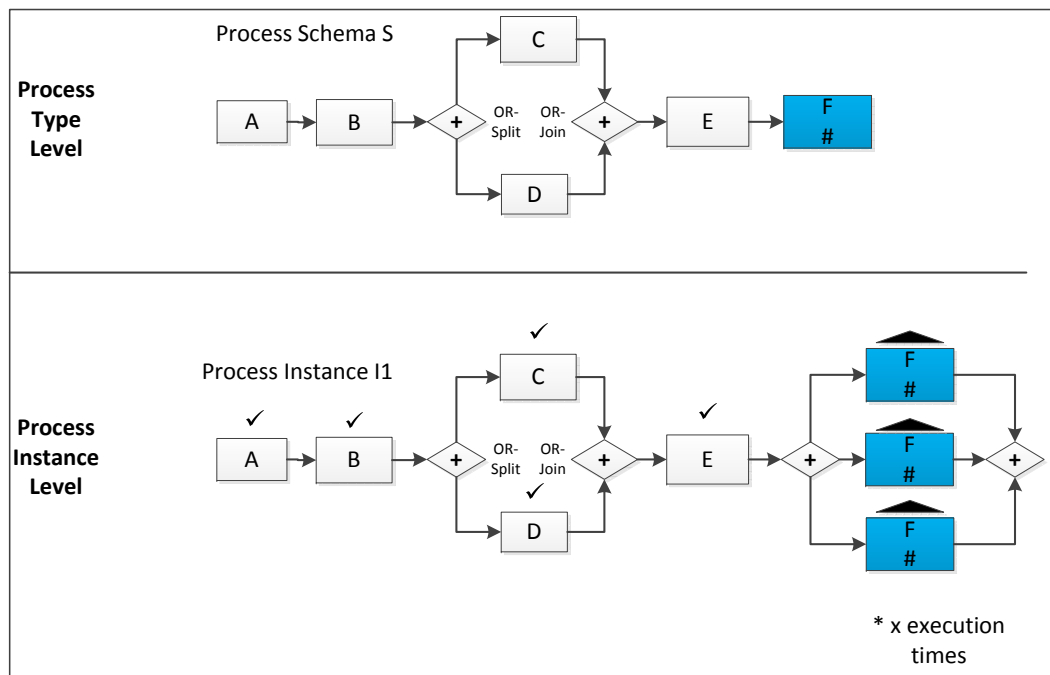


Abbildung 7.19: PP4: Multi-Instance Activity

7 Change Patterns Anwendungsbeispiel

8 Implementierung

In diesem Kapitel wird die Implementierung der Change Patterns Animationen beschrieben. Die Animation der Change Patterns hat das Ziel, das Konzept der Muster besser zu veranschaulichen und Flash testen zu können. Wenn man das Konzept in einem konkreten Beispiel sieht, kann man sich mit der Theorie vertraut machen.

8.1 Animationsübersichtsseite

Durch die Hauptseite kann man zu allen Webseiten des Projekts gelangen. Die Hauptseite wurde mit HTML5 implementiert und das Aussehen der Webseite mit CSS formatiert (siehe Abbildung 8.1).

Zur besseren Strukturierung der Seite habe ich Tabellen verwendet, um die Seitenlinks übersichtlicher zu halten. Die Links sind in blau gehalten und wenn man auf den Link klickt, wird die Farbe automatisch zu violett geändert. Die Farbänderung zeigt dem Nutzer, dass die Seite schon besucht wurde. Beispiel für eine Linkadresse:

```
<a href="apl_1.html">Insert Process Fragment</a> (siehe Abbildung 8.1).
```

8.2 Struktur der Animationsübersichtsseite

Das folgende Bild erläutert die verschiedenen Komponenten der implementierten Webseite (siehe Abbildung 8.2) für die Übersicht der Patterns.

8.2.1 Header

Der Header enthält das Logo und den Namen der Universität und den Namen der Abteilung, an der die Patterns entwickelt wurden.

8 Implementierung

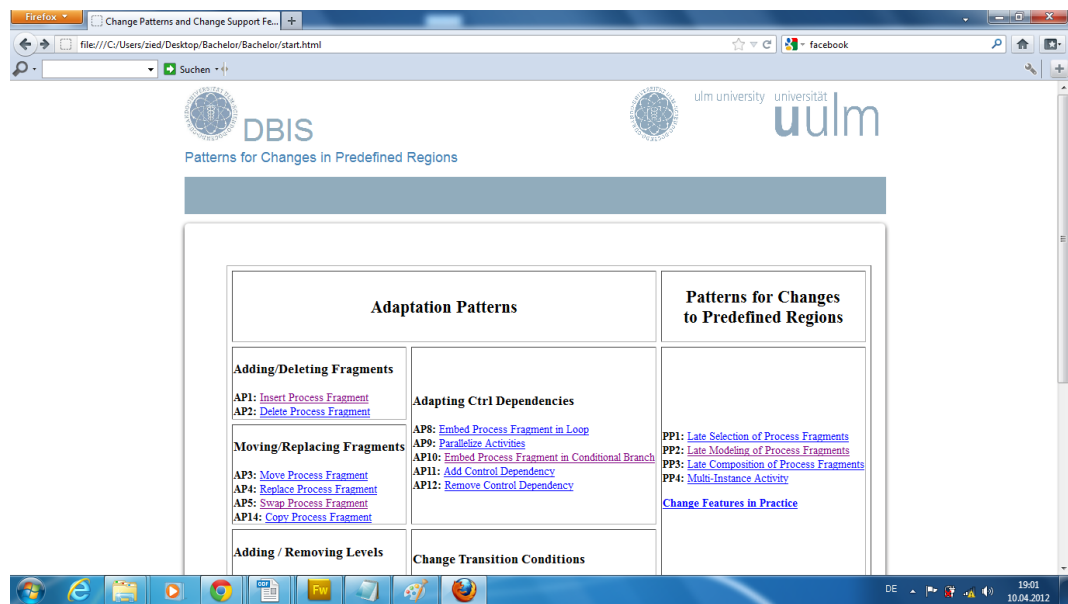


Abbildung 8.1: Animationsübersichtsseite

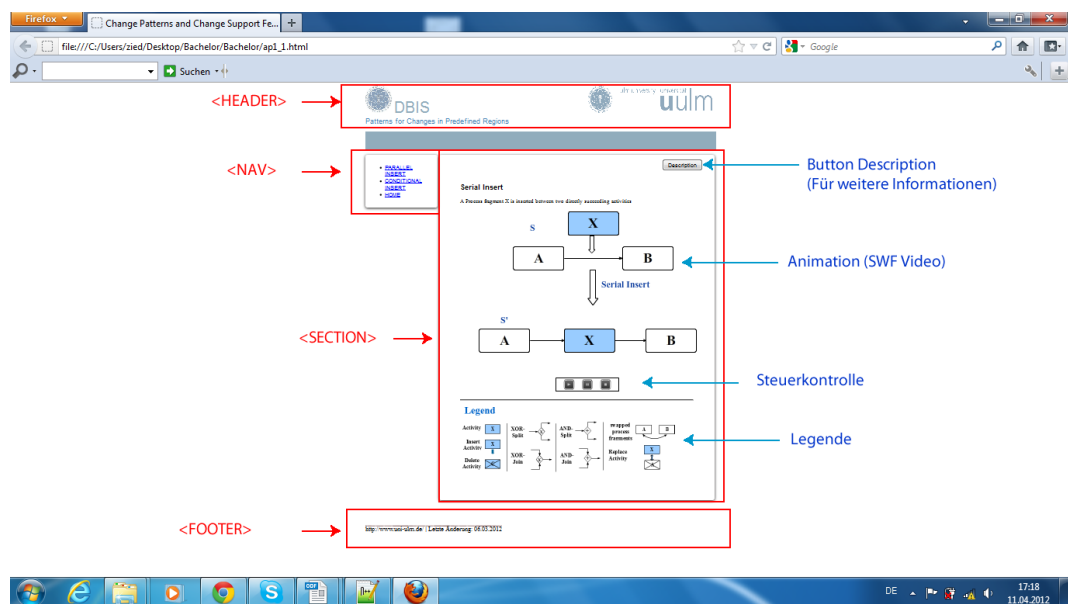


Abbildung 8.2: Struktur der Animationsübersichtsseite

8.2.2 Navigator

Der Navigator enthält das Menü, mit dem man entweder zur nächsten Seite gehen oder auf die Hauptseite zurückgehen kann.

8.2.3 Sektion

Sektion ist der Hauptbereich der Webseite, welcher die Animationen, die Legende, und den *Button Description* enthält.

8.2.4 Legende

Die Legende erläutert die Bedeutung der einzelnen Objekte, die in der Implementierung verwendet werden.

8.2.5 Steuerkontrolle

Die Steuerkontrolle dient dazu, Animationen anhalten, pausieren oder erneut abspielen zu können. Die Steuerkontrolle ist durch 3 Buttons realisiert (siehe Abbildung 8.3). Flash stellt dem Entwickler verschiedene vordefinierte Komponenten für Videos, wie zum Beispiel „PauseButton“, „StopButton“ oder „PlayButton“ usw., zur Verfügung. Diese vordefinierte Komponenten erleichtern die Entwicklung, das heißt der Programmierer muss solche Komponenten nicht selber programmieren, er kann diese Komponente daher direkt verwenden.



Abbildung 8.3: Steuerkontrolle

Die Steuerung dieser Komponente wird durch ActionScript3 realisiert. Flash stellt Actionscript Bibliotheken [40] zur Verfügung, ähnlich wie die anderen Programmiersprachen, beispielsweise die Java API. Ein Aufruf von Bibliotheken geschieht über das Schlüsselwort „import“. In meiner Arbeit habe ich die Bibliothek „MouseEvent“ benötigt.

Das Event "MouseEvent" entsteht, wenn der Anwender auf den Button drückt oder loslässt. Dieses Event beschreibt, wie das Programm auf einen Klick reagieren sollte.

8.2.6 Eventsyntax

Man kann ein Event auf eine Komponente wie folgt definieren (siehe Listings 8.1):

```
1 componentInstance.addEventListener(event, listener);
```

Listings 8.1: Eventsyntax

Listener: referenziert ein Listener Objekt oder eine Funktion.

Event: zeigt den Eventnamen an.

8.2.7 Funktionssyntax

Die Syntax für die Funktion ist wie folgt definiert (siehe Listings 8.2):

```
1 function functionName (param1, param2, etc...):void {  
2   Anweisungen;  
3 }
```

Listings 8.2: Funktionssyntax

8.2.8 Code für die Button zur Animationskontrolle

Hier ist der Code für die Steuerkontrolle dargestellt (siehe Listings 8.3):

```
1 import flash.events.MouseEvent;  
2  
3 play_button.addEventListener(MouseEvent.CLICK, play_button_onMouseclick);  
4 function play_button_onMouseclick(e:MouseEvent):void  
5 {  
6     play();  
7 }  
8  
9 stop_button.addEventListener(MouseEvent.CLICK, stop_button_onMouseclick);  
10 function stop_button_onMouseclick(e:MouseEvent):void  
11 {  
12     stop();  
13 }  
14 pause_button.addEventListener(MouseEvent.CLICK, pause_button_onMouseclick);  
15 function pause_button_onMouseclick(e:MouseEvent):void  
16 {  
17     stop();  
18 }
```

Listings 8.3: Code für den Play und Stop Button

8.2.9 Button „Description“

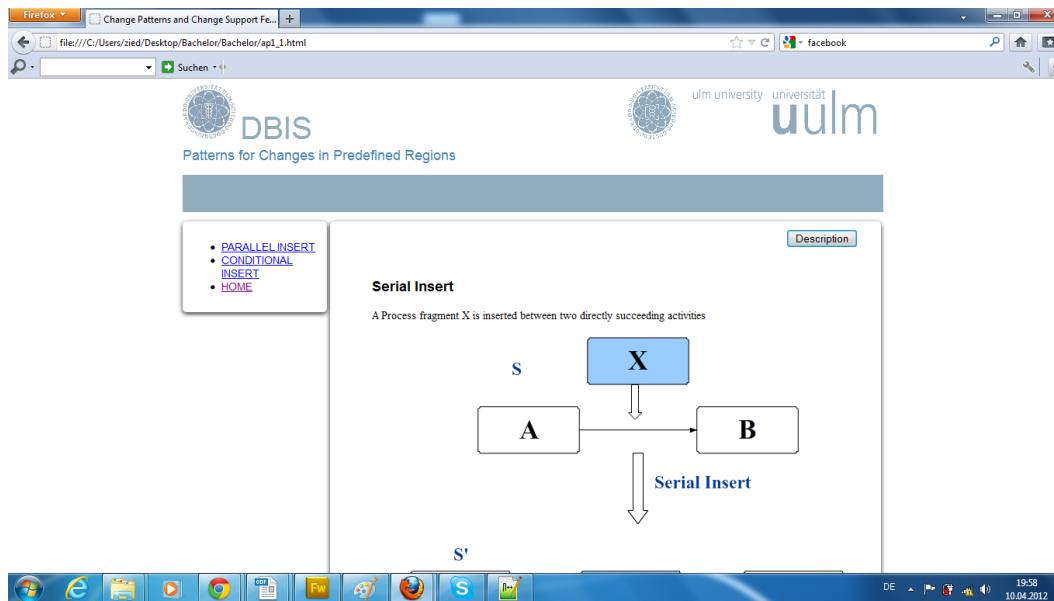


Abbildung 8.4: Serial Insert

Wenn man auf den Button „Description“ klickt (siehe Abbildung 8.4), gelangt man zu einer Informationsseite. Auf dieser Seite bekommt man alle Informationen über das Pattern angezeigt.

Der Button ist mit Javascript realisiert. Der Javascript Code wird in den HTML5 Code im Header genau zwischen dem `<\title>` und dem `<\head>` Tag eingefügt (siehe Listings 8.4).

```

1 <script type="text/javascript">
2     function FensterOeffnen () {
3         Fenster = window.open("AP1_1Des.html", "Zweitfenster1",
4                               "width=600,height=400");
5         if (Fenster.locationbar) {
6             if (Fenster.locationbar.visible == false) {
7                 Fenster.close();
8                 Neufenster = window.open("AP1_1Des.html", "Zweitfenster2",
9                                           "width=600,height=400,location=yes");
10                Neufenster.focus();
11            }
12        } else {
13            alert("Fehler aufgetreten!");
14        }
15    }
16 </script>

```

Listings 8.4: Button mit Javascript

8.2.10 Textanimation

Mithilfe der Werkzeugpalette erzeugt man einen Text. Aus diesem Text wird ein Objekt erzeugt. Flash bietet für Programmierer eine Bibliothek, die verschiedene bereits vorhandene Animationen enthält (siehe Abbildung 8.5, 8.6). Man kann sich diese Bibliothek zunutze machen, um die Texte animieren zu können.

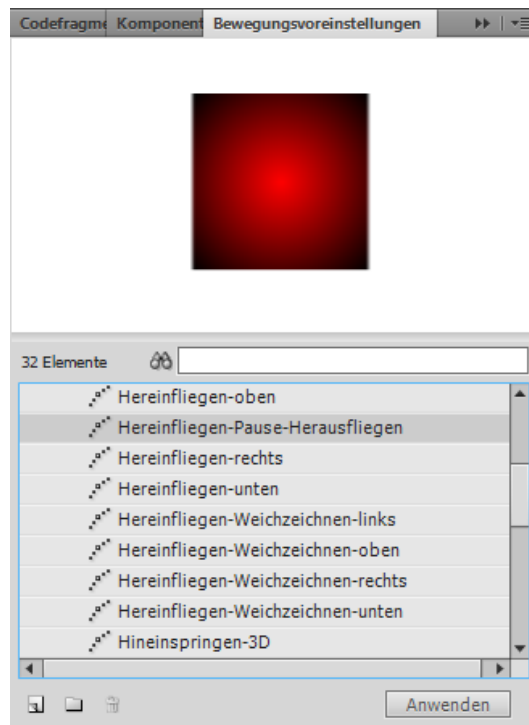


Abbildung 8.5: Bibliothek für Animationen

Die Einstellung der Bildrate ist sehr wichtig im Kontext von Animationen. Die Einstellung kann daher die Leistung des Computers stark beeinflussen. Eine zu hohe Bildrate kann zu Prozessorproblemen führen. Eine niedrige Bildrate führt zu einer langsamen Animation. Eine Bildrate mit 24 Bildern pro Sekunde ist optimal und erzeugt eine schnelle und glatte Animation (siehe Abbildung 8.7).

8.2.11 Hauptanimation

Die komplexen Animationen werden stückweise animiert, das heißt sie werden in verschiedene Komponenten unterteilt und jede Komponente wird in ein Objekt konvertiert. Jedes



Abbildung 8.6: Animierter Text

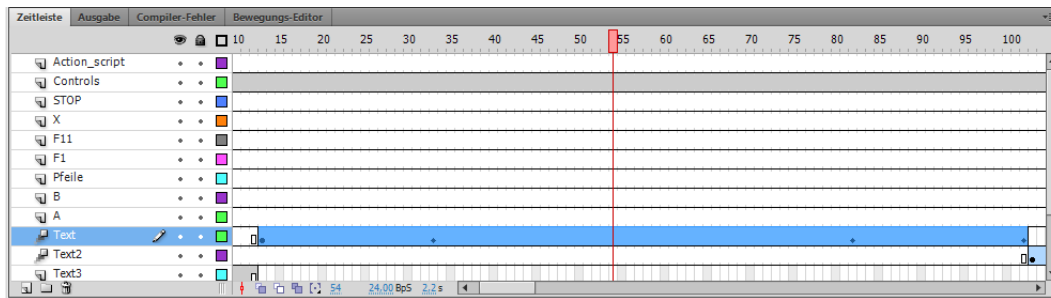


Abbildung 8.7: Zeitleiste + Bildrate

Objekt wird auf einer separaten Ebene bearbeitet und animiert. Wie auf der Abbildung 8.7 ersichtlich ist, ist die Zeitleiste in verschiedene Ebenen unterteilt. In Abbildung 8.8 werden mit den roten Pfeilen ein paar Beispiele für Komponenten und die jeweilige Ebenen, auf denen die Objekt animiert wurden, gezeigt. Als Beispiel ist das Objekt „Node A“ durch verschiedene Komponenten (Text, Rechteck) zusammengefasst. Das gesamte Zeichen (Text, Rechteck) wird in ein einziges Objekt konvertiert. Basierend auf klassischen Tweens wird „Node A“ animiert. Die Komponenten des Objektes „Node A“ können auch bearbeitet werden. Durch einen Doppelklick auf das Objekt „Node A“ kann man auf die Komponente einzeln zugreifen und diese bearbeiten. Auf der Menüleiste von Flash wird angezeigt, dass wir gerade innerhalb von einem Objekt stehen und die jeweiligen Komponenten bearbeiten können (siehe Abbildung 8.9).

8.2.12 Animationen in HTML5 integrieren

Die Animationen werden als SWF-Film exportiert. Diese SWF-Filme können in einer Webseite eingebettet werden (siehe Abbildung 8.10).

Den SWF-Film kann man dann mit folgendem Code in eine Webseite einbetten (siehe Listings 8.5):

8 Implementierung

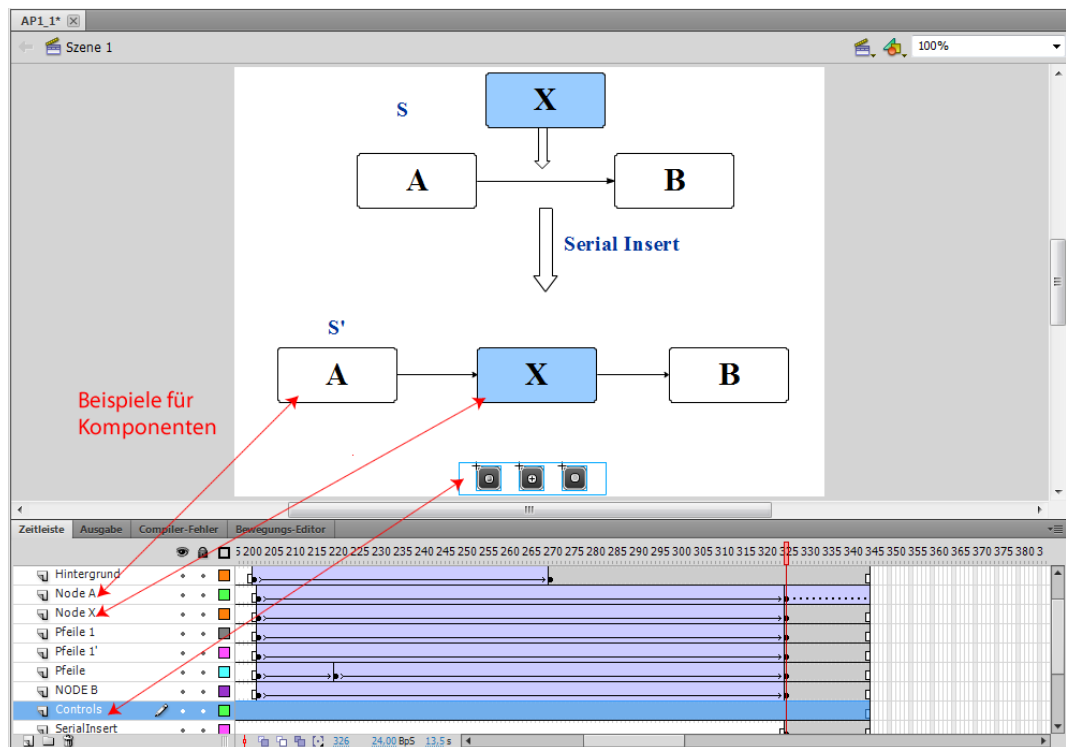


Abbildung 8.8: Bearbeitung von Objekten auf einer Ebene

8.3 Abgleich der gestellten Anforderungen

In dieser Arbeit wurden die auf Kapitel 2.3 genannten Anforderungen berücksichtigt. Die Process Change Patterns wurden alle erfolgreich animiert. Schließlich sind diese Animationen in eine HTML5-Webseite integriert (siehe Abbildung 8.11).

8.3 Abgleich der gestellten Anforderungen

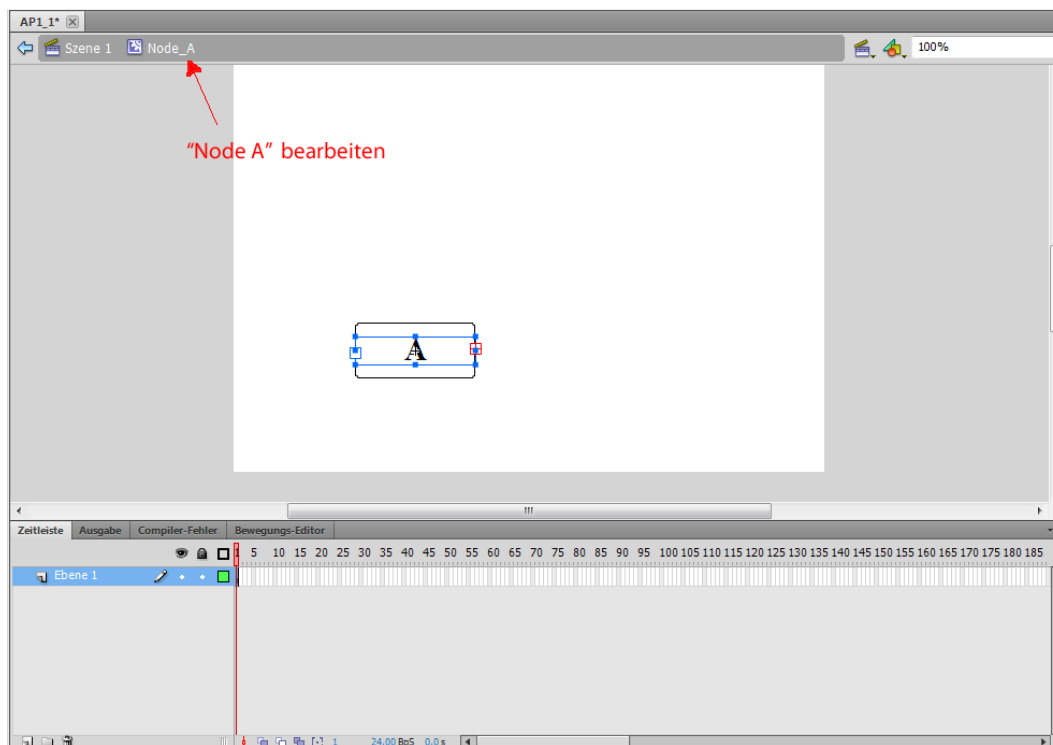


Abbildung 8.9: Objekte bearbeiten

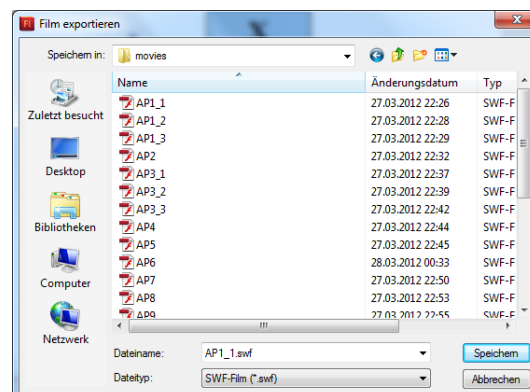


Abbildung 8.10: SWF exportieren

```
1 <object width="650" height="470" type='application/x-shockwave-flash' data='movies/PP3.swf' width='400' height='400'>
2   <param name='movie' value='movies/PP3.swf' />
3   <param name='allowfullscreen' value='true' />
4   <param name='allowsriptaccess' value='always' />
5   <param name='flashvars' value='file=../clips/video.flv&image=flvplayer/preview.jpg' />
6   <p><a href='http://get.adobe.com/flashplayer'>Get Flash</a> to see this player.</p>
7 </object>
```

Listings 8.5: SWF-Film in eine Webseite einbetten

Abgleich der gestellten Anforderungen (siehe dazu auch Kapitel 2.3)

Muster	Implementiert	Bewertung
Adaptation Patterns		
Pattern AP1: Insert Process Fragment (Serial Insert)	ja	*****
Pattern AP1: Insert Process Fragment (Parallel Insert)	ja	*****
Pattern AP1: Insert Process Fragment (Conditional Insert)	ja	*****
Pattern AP2: Delete Process Fragment	ja	*****
Pattern AP3: Move Process Fragment (Serial Move)	ja	*****
Pattern AP3: Move Process Fragment (Parallel Move)	ja	*****
Pattern AP3: Move Process Fragment (Conditional Move)	ja	*****
Pattern AP4: Replace Process Fragment	ja	*****
Pattern AP5: Swap Process Fragment	ja	*****
Pattern AP6: Extract Process Fragment to Sub Process	ja	*****
Pattern AP7: Inline Sub Process	ja	*****
Pattern AP8: Embed Process Fragment in Loop	ja	*****
Pattern AP9: Parallelize Process Fragments	ja	*****
Pattern AP10: Embed Process Fragment in Conditional Branch	ja	*****
Pattern AP11: Add Control Dependency	ja	*****
Pattern AP12: Remove Control Dependency	ja	*****
Pattern AP13: Update Condition	ja	*****
Pattern AP14: Copy Process Fragment	ja	*****
Patterns for Changes in Predefined Regions		
Pattern PP1: Late Selection of Process Fragments	ja	*****
Pattern PP2: Late Modeling of Process Fragments	ja	*****
Pattern PP3: Late Composition of Process Fragments	ja	*****
Pattern PP4: Multi Instance Activity	ja	*****

Abbildung 8.11: Abgleich der gestellten Anforderungen

9 Zusammenfassung

scrpage2 []Vladimir []Individuelles Gestalten von Kopfzeilen [55]55 In der vorgestellten Arbeit wurden verschiedene Techniken wie Flash, HTML5, CSS und Javascript vorgestellt und verglichen, anschließend wurden diese in einem konkreten Beispiel umgesetzt, nämlich im Kontext der Animationen der „Change Patterns“. Die Animationen können angehalten, pausiert oder erneut abgespielt werden. Abschließend wurden die Animationen auf einer Webseite veröffentlicht, um die Animationen besser betrachten zu können.

Ein Vorteil von HTML5 ist, dass man für Videos keine extra Plug-ins installieren muss, aber leider existieren ausgereifte Editoren für HTML5 Animationen bis heute nur in Beta Versionen. Für kleine Animationen muss man einen großen Programmieraufwand betreiben. Die meisten Videos im Internet laufen bis jetzt auf der Basis von Flash. Wie lange Flashvideos im Internet noch genutzt werden, bis sie durch HTML5 Videos vollständig ersetzt werden, dass wird die Zukunft bringen müssen.

Literaturverzeichnis

- [1] Ressourcen für Flash CS3. <http://www.adobe.com/support/documentation/de/flash/>. Last visited at 19. November 2012
- [2] Mediaplayer. <http://de.wikipedia.org/wiki/Media-Player>. (Last visited at 2012)
- [3] Quick Time. <http://de.wikipedia.org/wiki/QuickTime>. (Last visited at November 2012)
- [4] Realplayer. <http://de.wikipedia.org/wiki/Realplayer>. (Last visited at October 2012)
- [5] Macromedia. <http://en.wikipedia.org/wiki/Macromedia>. (Last visited at November 2012)
- [6] Youtube. <http://de.wikipedia.org/wiki/YouTube>. (Last visited at December 2012)
- [7] Dailymotion. <http://en.wikipedia.org/wiki/Dailymotion>. (Last visited at November 2012)
- [8] Apple Inc. http://en.wikipedia.org/wiki/Apple_Inc. (Last visited at November 2012)
- [9] Adobe Dreamweaver. http://de.wikipedia.org/wiki/Adobe_Dreamweaver. (Last visited at November 2012)
- [10] Adobe Muse. <http://muse.adobe.com>. (Last visited at 2012)
- [11] Adobe AIR 3. <http://www.adobe.com/de/products/air.html>. (Last visited at 2012)
- [12] Adobe Wallaby. http://en.wikipedia.org/wiki/Adobe_Wallaby. (Last visited November 2012)
- [13] Weber, B., Reichert, M., Rinderle-Ma, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering. Elsevier Science, 2008, 66(3): 438-466.
- [14] Weber, B., Rinderle, S., Reichert, M.: Process Change Patterns (Aktuelles Schlagwort). EMISA Forum, 2007, 27(2): 45-51.
- [15] Rinderle-Ma, S., Reichert, M., Weber, B.: On the Formal Semantics of Change Patterns in Process-aware Information Systems. Proc. 27th Int'l Conference on Conceptual Modeling (ER'08), Springer, 2008, LNCS 5231, pp. 279-293.
- [16] Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. Proc. 11th Int'l Conf. on Advanced Information Systems Engineering (CAiSE'07), Springer, 2007, LNCS 4495, pp. 574-588.
- [17] Weber, B., Lanz, A., Reichert, M.: Time Patterns for Process-aware Information Systems - A Pattern-based Analysis. University of Ulm, Faculty of Electrical Engineering and Computer Science, 2009, Report No. UIB-2009-05.
- [18] Pilgirm, M.: Up and Running. O'Reilly Media, Inc, USA, 2010.
- [19] Kröner, P.: HTML5. Webseiten innovativ und zukunftssicher. Open Source Press, 2011.

Literaturverzeichnis

- [20] Kilgenstein, J.: CSS einfach erklärt - Eine Einführung ins Webdesign anhand konkreter Beispiele, Kretschmann Mediamarketing, 2009.
- [21] Bruce Lawson, R. S.: HTML5. Eine Einführung für Umsteiger. Addison-Wesley, 2011.
- [22] Brian P., Hogan, S. F.: HTML5 & CSS3: Webentwicklung mit den Standards von morgen. O Reilly, 2011.
- [23] Nick Weschkalnies, S. G.: Adobe Flash CS5: Das umfassende Handbuch (Galileo Design), Galileo Design, 2010.
- [24] APPS und Webapps - Definition. <http://www.app-entwickler-verzeichnis.de/faq-app-entwicklung/11-definitionen/31-apps-und-webapps-definition>. (Last visited at October 2012)
- [25] Tables. <http://www.w3.org/TR/html4/struct/tables.html>.
- [26] Flash CS4 Professional ActionScript 2.0. http://help.adobe.com/en_US/AS2LCR/Flash_10.0/help.html?content=00002328.html. (Last visited at 2012)
- [27] ActionScript 3. http://help.adobe.com/de_DE/FlashPlatform/reference/actionscript/3/index.html. (Last visited at November 2012)
- [28] Flash-Hilfe. http://help.adobe.com/de_DE/Flash/10.0_UsingFlash/WSd60f23110762d6b883b18f10cb1fe1af6-7b1aa.html. (Last visited at 2012)
- [29] Webentwicklung: HTML5 vs Flash – Wer gewinnt das Duell? <http://t3n.de/news/html5-flash-duell-268241/>. (Last visited at 2012)
- [30] HTML5 Das Web von morgen. http://www.chip.de/artikel/HTML5-Das-Web-von-morgen_41539437.html. (Last visited at November 2012)
- [31] Flash stirbt: HTML 5 verändert Adobe's Plug-ins. http://www.chip.de/artikel/HTML5-Das-Web-von-morgen-6_41539577.html. (Last visited at 2012)
- [32] HTML5: Das Web von Morgen – Webdesign der nächsten Generation. <http://pc-pedia.de/2011/07/html5-das-web-von-morgen-webdesign-der-nachsten-generation/>. (Last visited at Juli 2011)
- [33] WebKit. <http://en.wikipedia.org/wiki/WebKit>. Last visited at 19. November 2012
- [34] Hilfe zu Flash Professional: <http://helpx.adobe.com/de/flash/using/motion-tween-animation.html>. (Last visited 2012)
- [35] Thoughts on Flash. <http://www.apple.com/hotnews/thoughts-on-flash/>. (Last visited at 2012)
- [36] Digital rights management. http://en.wikipedia.org/wiki/Digital_rights_management. (Last visited at December 2012)
- [37] Digital Rights Management. <http://www.interlake.net/de/interlake-media/digital-rights-management/>.
- [38] HTML Reference. http://www.w3schools.com/html5/tag_canvas.asp. (Last visited at 2012)

- [39] Adobe Edge Animate. <http://labs.adobe.com/technologies/edge/>. (Last visited at 2012)
- [40] MouseEvent -AS3. http://help.adobe.com/de_DE/FlashPlatform/reference/actionscript/3/flash/events/MouseEvent.html. (Last visited at November 2012)
- [41] HTML5 Canvas Linear Motion Animation Tutorial. <http://www.html5canvastutorials.com/advanced/html5-canvas-linear-motion-animation/>. (Last visited at November 2012)
- [42] Kreher, U., Reichert, M., Rinderle-Ma, S., Dadam, P.: Effiziente Repräsentation von Vorlagen- und Instanzdaten in Prozess-Management-Systemen. Technical Report. University of Ulm, 2009, Report No. UIB-2009-08.
- [43] Getting to Grips with HTML5 Browser Compatibility. <http://speckyboy.com/2012/03/25/getting-to-grips-with-html5-browser-compatibility/>. (Last visited at 2012)
- [44] Robecke, A.: Development of an iPhone business application. University of Ulm, 2011.
- [45] Schmid, M.: Mobile Web Service Implementierung am Beispiel des iPhone OS 4.0. University of Ulm, 2010.
- [46] Weber, B., Rinderle-Ma, S., Reichert, M.: Change Support in Process-Aware Information Systems - A Pattern-Based Analysis. University of Twente, 2007, Technical Report, TR-CTIT-07-76.
- [47] Weber, B., Rinderle, S., Reichert, M.: Identifying and Evaluating Change Patterns and Change Support Features in Process-Aware Information Systems. University of Twente, 2007, Technical Report, TR-CTIT-07-22.
- [48] Weber, B., Sadiq, S., Reichert, M.: Beyond Rigidity - Dynamic Process Lifecycle Support: A Survey on Dynamic Changes in Process-aware Information Systems. Computer Science - Research and Development, Springer, 2009, 23(2): 47-65.
- [49] Reichert, M., Weber, B., Rinderle, S.: Process Change Support Features (Aktuelles Schlagwort). EMISA Forum, 2008, 28(1): 24-30.
- [50] Li, C., Reichert, M., Wombacher, A.: On Measuring Process Model Similarity based on High-level Change Operations. Proc. 27th International Conference on Conceptual Modeling (ER'08), Springer, 2008, LNCS5231, pp. 248-264.

Name: Zied Khzami

Matrikelnummer: 613391

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Zied Khzami