

# Grafische Methoden auf dem Web

## — Eine Übersicht

Wolfgang Leister      Arve Larsen

Norsk Regnesentral, Oslo\*

### Zusammenfassung

Es zeichnet sich ab, daß sich das WWW zur kommenden Plattform für Interaktion mit dem Benutzer entwickelt. Das hier beschriebene Projekt untersucht die Möglichkeiten, grafische Benutzeroberflächen auf das WWW abzubilden. Bildformate, CGI-Techniken, Java, CORBA, VRML, Plug-In-Technologien, usw. werden in einer Übersicht zusammengefaßt. Gleichzeitig wird untersucht, wie ältere Grafikanwendungen mit einer WWW-Schnittstelle ausgerüstet werden können, was anhand des Grafik-Paketes GPGS der NORSIGD erläutert wird.

## 1 Wieso diese Übersicht?

Die Technologie des Web wird nicht nur zum reinen Informationszugang benutzt. Viele Anwendungen, auch grafische, werden mit Hilfe der Web-Technologie bedient. Die Benutzerschnittstelle des Web entwickelt sich zu einem anwendungsübergreifenden Standard. Dabei entsteht eine strikte Trennung zwischen Darstellung und Anwendungslogik. Dadurch können Ressourcen an geografisch verschiedenen Orten aufgebaut sein, während der Benutzer alle wichtigen Daten von seinem Arbeitsplatz aus zugreifen kann.

Viele bestehende Anwendungen sind noch nicht an das neue Medium angepaßt. Speziell für Anwendungen, die grafische Ressourcen benötigen, müssen beim Übergang die Verhältnisse in Bezug auf die Systemarchitektur, den Ressourcenverbrauch, und andere Gegebenheiten auf dem WWW berücksichtigt werden. Oft sind Berechnungs-, Simulations- oder Visualisierungsanwendungen, die teilweise auf älteren Grafikbibliotheken aufgebaut sind, schon jahrelang ohne große Veränderungen in Betrieb. Beim Übergang auf ein neues Medium ist es zweckmäßig, nicht alles radikal zu verändern, sondern soviel wie möglich wiederzuverwenden.

NORSIGD (Norwegischer Verein für Computergrafik) hat die grafische Bibliothek GPGS seit der Mitte der siebziger Jahre entwickelt und gewartet. Diese Untersuchung soll eine Antwort darauf geben, wie Anwendungen, die diese Bibliothek benutzen, für den Gebrauch im WWW angepaßt werden können<sup>1</sup>. Gleichzeitig sollen diese Erfahrungen auch für andere Bibliotheken und bei der Neuentwicklung von Anwendungen nützlich sein.

---

\*Norsk Regnesentral, Postboks 114 Blindern, N-0314 Oslo, Norwegen, email: Wolfgang.Leister@NR.no

<sup>1</sup>Dieser Bericht über grafische Techniken auf dem WWW wurde von der NORSIGD in Auftrag gegeben und finanziert. Die vorliegende Ausarbeitung ist eine Zusammenfassung dieser Arbeit.

Dieser Bericht hat nicht zum Ziel ein Handbuch zu sein, oder alle bekannten Möglichkeiten aufzulisten. Problemstellungen, die sich auf Multimedia oder auf digitales Fernsehen beziehen, sind nicht Teil dieses Berichts, obwohl diese Gebiete sich auf einige Aspekte der Computergrafik und des WWW beziehen. Innerhalb der Multimedientechnik steht die Synchronisierung und Integration verschiedener Datentypen im Vordergrund. Innerhalb des digitalen Fernsehens beschäftigt man sich eher mit Streaming-Techniken und der Bereitstellung von Inhalt in einer Rundfunk-Metapher. Dieser Bericht soll auch nicht den Aufbau des Internet als solches (z.B. Protokolle der Transportschicht, Fragen zur Dienstgüte etc.) behandeln.

## 2 Aufbau des WWW

World Wide Web, WWW, oder auch W3 wurde von Tim Berners-Lee während eines Forschungsaufenthaltes bei CERN vorgeschlagen. Im Herbst 1980 entwickelte er ein Programm mit dem Namen "Enquire-Within-Upon-Everything", das es möglich machte in Dokumente Verweise auf zufällig gewählten Knoten einzufügen.<sup>2</sup>

Im Jahre 1989 schlug Berners-Lee ein weltumspannendes Hypertext-Projekt vor, das sich mit der Zeit zum WWW entwickeln sollte. WWW wurde unter der Vorgabe entwickelt, daß mehrere Personen zusammenarbeiten, indem sie Information in einem Gewebe aus Hypertext-Dokumenten ablegen. WWW-Dokumente werden auf einer Wirtsmaschine abgelegt, während ein Client-Programm, das Browser genannt wurde, den Zugang zu der bereitgestellten Information verschafft. Der Browser ermittelt die Information indem er dem URL eines Verweises folgt und das Dokument mit dem HTTP-Protokoll überträgt.

WWW ist ein Dienst, der ursprünglich zum Betrieb auf dem Internet entwickelt wurde, um Benutzergruppen und Anwendungen den Zugang auf Dokumente und Dienste bereitzustellen. Damit ist WWW ein Teil der Middleware. WWW macht Dokumente auf dem Internet zugänglich und hat sich zu einem globalen Informationssystem entwickelt. Mit dem WWW kann eine Standardisierung der Benutzeroberfläche erreicht werden. Die Endbenutzer können alle Anwendungen mit derselben Benutzeroberfläche über Firmen-, Industrie- oder Landesgrenzen hinweg ausführen.

Dem WWW sind folgende Technologien zugrundegelegt:

- Web server
- Web browser
- Uniform Resource Locator (URL)
- Hypertext Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Web Navigation und Suchmaschinen
- Übergangspunkte (Gateways) um auf Nicht-WWW-Ressourcen zuzugreifen.

---

<sup>2</sup>Das System wurde zunächst auf Maschinen von Norsk Data unter dem Betriebssystem SINTRAN-III implementiert.

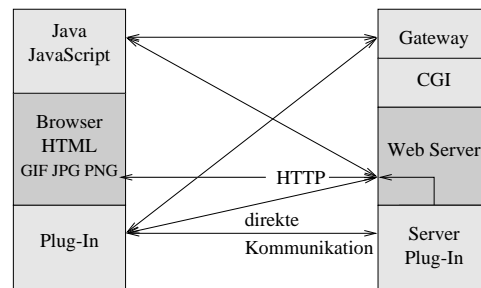


Abbildung 1: Komponenten des WWW

## 2.1 Browser

Ein Browser ist die Benutzeroberfläche zum WWW, die einfach in der Bedienung sein sollte. Mit Hilfe von u.a. Pull-Down-Menüs, Pop-Up-Menüs, Tasten und Kommandos werden der Zugang zu einem anderen Server gegeben, die Möglichkeit Dokumente zu lesen, Resultate auszudrucken, Programme auszuführen, Dokumente zu speichern und im WWW zu navigieren. Es gibt eine Vielzahl verschiedener Browser-Realisierungen, u.a. Lynx, Mosaic, Netscape Navigator/Communicator, Internet Explorer, Arena, Amaya und Opera.

Normalerweise werden Daten nur auf Benutzeraufforderung in den Browser übertragen. Im Gegensatz dazu ist **Push technology** eine Softwarekomponente, die entweder Information direkt an einen Rechner liefert, ohne daß diese Information direkt bestellt wurde, oder die informiert, wenn Dokumente bereitgestellt wurden oder in einer neueren Version vorhanden sind.

## 2.2 HTTP

HTTP (HyperText Transfer Protocol) ist ein Protokoll auf Anwendungsebene, das für das WWW entwickelt wurde. HTTP baut eine Verbindung zu einem Webserver auf und überträgt Dokumente zum Browser in vier Schritten: Verbindungsaufbau, Anfrage, Antwort und Verbindungsabbau. Es ist für die weiteren Betrachtungen wichtig, daß das Protokoll keine Speicherung der Statusinformation vorsieht. Jede Anfrage wird getrennt von der nächsten durchgeführt.

Die übertragenen Dokumente können auch Referenzen auf andere Objekte enthalten, die in das Dokument integriert sind. Jedes dieser Objekte wird über eine eigene HTTP-Anfrage übertragen.

## 2.3 HTML

HTML (Hyper Text Meta Language) ist eine einfache, plattformunabhängige Sprache um Web-Dokumente zu beschreiben. HTML unterstützt unter anderem Hypertext-Verweise als

URL in einem Dokument. Dokumente können auch andere Datentypen enthalten wie beispielsweise Grafik, Multimedia, Audio oder Video.

HTML baut auf SGML (Standard Generalized Markup Language) auf, die Dokumente über Plattformgrenzen hinweg zugänglich machen soll. HTML-Dokumente werden in einer Text-Datei abgelegt und mit Hilfe einiger Spezialcodes beschrieben. Zur Formatierung werden HTML-Tags verwendet, um Überschriften, Kursiv- und Fettschriften, Listen, Tabellen, Grafik, und weitere Objekte zu beschreiben. Die Möglichkeiten das Layout einer Seite exakt anzugeben, oder gar grafische Objekte zu benutzen, sind äußerst begrenzt. Wird eine genaue Wiedergabe eines Dokuments gewünscht, werden Style-Sheets oder Plug-Ins für spezielle Formate (z.B. Acrobat Reader) eingesetzt.

HTML bietet auch die Möglichkeit strukturierte Formulare aufzubauen. Diese werden in eine Webseite integriert, damit ein Benutzer Daten in die Eingabefelder eingeben oder aus Menüpunkten auswählen kann. Diese Daten werden zusammen mit der HTTP-Anfrage über den Server zur weiteren Bearbeitung an eine Web-Anwendung weitergeleitet.

**Dynamic HTML** wurde vom WWW Konsortium (W3C) definiert. Es ist eine Kombination aus Style Sheets und Scripts, die es ermöglichen, ein Dokument dynamisch zu ändern. Damit können Web-Autoren Objekte steuern, indem spezielle Webseiten-Elemente und HTML-Objekte geändert werden. Eine Objektorientierte Skriptsprache ermöglicht es, die Attribute von Grafik und Text unabhängig voneinander zu verändern.

## 2.4 Einfache Grafik auf dem WWW

Rastergrafiken und Bilder werden gewöhnlicherweise mit dem IMG-Tag in ein HTML-Dokument eingefügt. Normalerweise werden die Formate GIF und JPEG eingesetzt, während das Format PNG in der Zukunft der Standard werden kann. Formate, die im Browser nicht direkt implementiert sind, können durch ein Browser-Plug-In abgearbeitet werden.

Einige der Formate (z.B. GIF) gestatten es Animationen in Form einer Folge von Einzelbildern zu kodieren, um einfache, im Voraus berechnete Animationen vorzuführen zu können. Mit Hilfe des MIME-Multipart-Tags ist es in einem gewissen Maße möglich einen Medienstrom zu präsentieren, jedoch ohne Möglichkeit die Dienstgüte zu steuern.

Ein **Image Map** ist die Definition von geometrischen 2D-Bereichen, die einem Bild unterlegt werden. Diese Bereiche werden einem URL zugeordnet, das aktiviert wird, wenn man den entsprechenden Bereich anklickt. Damit können Benutzerschnittstellen entwickelt werden, bei denen Rasterbilder mit einem Image-Map unterlegt werden. Die Abarbeitung einer solchen Benutzerschnittstelle kann auf dem Browser (z.B. in JavaScript) oder auf dem Server (z.B. mit Hilfe der CGI-Schnittstelle) durchgeführt werden.

## 2.5 Externe Anwendungen und Browser-Plug-In

Die Funktionalität eines Browsers kann durch externe Anwendungen und Programme (Helper Applications) und durch Browser-Plug-Ins erweitert werden. Dadurch können Datentypen zum Abspielen von Ton, Video, Animationen oder Grafik unterstützt werden.

Für externe Anwendungen wird eine Verknüpfung zwischen dem MIME Typ und dem Name und den Parametern dieser Anwendung hergestellt, die zuvor auf dem entsprechenden Rechner installiert sein muß. Die Steuerung der Anwendung geschieht außerhalb der Verantwortung des Browsers. Es gibt gewisse Möglichkeiten für eine lose Kommunikation zwischen Anwendung und Browser (z.B. indem man den Browser oder die Anwendung mit speziellen Parametern aufruft).

Plug-ins sind Anwendungsmodule, die innerhalb des Browsers ausgeführt werden. Obwohl diese in der Ausführungsumgebung des Browsers in Bezug auf den Speicherbereich, Threads, Präsentation und andere Ressourcen ablaufen, sind Plug-Ins als eigenständige Module programmiert. Plug-Ins werden oft als dynamische Bibliothek implementiert, z.B. als .dll (Windows) oder .so (Unix) mit einem fest definierten API. Teilweise werden die Browser-Funktionen durch JavaScript, Java-Klassen oder ein C-Interface aufgerufen. Datenströme, Parameter und Handles von Fenstern auf dem Bildschirm werden beim Aufruf an das Plug-In weitergereicht und dort weiterverwendet. Implementierungsdetails für Plug-Ins für die verschiedenen Browser findet man auf den Web-Seiten für Entwickler, z.B. [Net98]. Man sollte auch beachten, daß Browser Plug-Ins plattformspezifisch sind. In heutigen Browsern verwendet man Plug-Ins beispielsweise um PNG-kodierte Bilder oder Video zu zeigen. Es ist auch denkbar, ältere Grafikbibliotheken (z.B. GPGS) und Anwendungen als Plug-In zu implementieren.

## 2.6 Web Server und Web Gateway

Ein **Web server** besteht aus einem Rechner und der dazugehörigen Software mit den folgenden drei Hauptfunktionen:

- abspeichern von Dokumenten, die in HTML kodiert sind, und anderes Material,
- ein Programm, das Anforderungen von einem Browser entgegennimmt und die entsprechenden Daten an den Browser sendet.
- Gateways, die Web-Inhalte (z.B HTML-Dokumente) erzeugen können und damit einen Zugang zu Nicht-Web-Dokumenten ermöglichen.

Ein **Web Gateway** ist ein Dienst, der einem Browser Zugang zu Daten verschafft, die ansonsten für das WWW nicht zugänglich wären. Beispiele dafür sind firmeninterne Anwendungen, Datenbanken, ältere Programmsysteme und auch grafische Anwendungen. Diese Gateways greifen auf die Information zu, konvertieren den Inhalt in ein geeignetes Format für den Browser und überträgt die Information mit dem HTTP-Protokoll. Zusätzlich zu der reinen Konvertierfunktion muß ein solches Gateway auch Statusinformationen hantieren, wie sie in traditionellen Benutzeroberflächen und Anwendungen benutzt werden. Neben verschiedenen Techniken auf der Client-Seite werden vor allem Common Gateway Interface (CGI), Server-Side Includes (SSI) und Standalone Server zur Implementierung benutzt.

## 2.7 Uniform Resource Locator (URL)

URL ist die Basis, um Ressourcen auf dem WWW adressieren zu können. Ein URL gibt das Protokoll, den Server, die Port-Nummer, den Pfad (Dateiname oder Dienst) und eventuelle

Parameter an. Für einige der Parameter gibt es Default-Werte, falls ein URL nur teilweise angegeben wird. URL werden sowohl in HTML-Dokumenten als auch in Browsern verwendet, um Ressourcen anzugeben. URL wird durch die IETF (Internet Engineering Task Force, <http://www.ietf.org/>) in RFC 1738 definiert.

## 2.8 MIME

Gemäß dem Internet-Standard RFC822 dürfen Email-Botschaften lediglich aus reinem Text bestehen. Mit MIME (Multipurpose Internet Mail Extensions, RFC 2045-2049) können unter anderem auch Daten übertragen werden, die auf anderen Zeichensätzen aufbauen, in einem binären Format vorliegen, oder aus mehreren Teilen bestehen.

MIME wurde außerhalb des eigentlichen Geltungsbereiches erfolgreich verwendet, um die Medientypen auf dem WWW festzulegen. MIME definiert unter anderem eine Versionsnummer, um geeignete Programmversionen zu ermitteln, den Inhaltstyp (content-type, siehe RFC 1049), und die Übertragungskodierung (Content-Transfer-Encoding).

Mit dem MIME-Typ *multipart* können mehrere Dokumente innerhalb eines Datenstroms untergebracht werden. Ursprünglich um mehrere Anlagen in elektronischen Briefen zu ermöglichen, wird dies in Browsern ausgenutzt, um Bildsequenzen zu übertragen. Die Bilder werden in der richtigen Reihenfolge getrennt durch den multipart Header gesendet und im Browser bei ihrer Ankunft angezeigt. Dadurch ist in gewissem Maße die Übertragung von Animationen möglich.

## 3 Rastergrafik-Formate

In diesem Abschnitt werden die am häufigsten auf dem WWW benutzten Grafikformate vorgestellt. GIF und jpeg sind Formate, die schon lange benutzt werden, während PNG ein Nachfolger der Formates GIF darstellt. Eine Einführung in Grafikformate auf dem WWW wird von W3C bereitgestellt [BLG97].

### 3.1 PNG

PNG (Portable Network Graphics) ist ein erweiterbares Dateiformat für die verlustfreie Komprimierung und portable Speicherung von Rasterbildern. Das Format wurde vor allem für den Gebrauch im WWW entwickelt, nachdem ein Patentstreit um das GIF Format entstanden war. PNG ersetzt in mancher Hinsicht auch das vielbenutzte TIFF Format. Es werden Bilder mit Farbtabelle, Grauwertbilder und True Color Bilder unterstützt. Zusätzlich kann auch ein Alpha-Kanal verwendet werden. Die Farbauflösung variiert zwischen 1 und 16 Bit pro Pixel und Kanal. PNG kann auch in einem Pixelstrom mit einer progressiven Anzeigeoption verwendet werden. PNG ist robust und kann auf gewöhnliche Übertragungsfehler hin überprüft werden.

## 3.2 JPEG

JPEG (wird wie “dschei-peg” ausgesprochen) ist ein standardisiertes Bildkomprimierungsverfahren. JPEG ist ein Akronym für *Joint Photographic Experts Group*, die den ISO-Standard entwickelt hat. JPEG eignet sich zur Komprimierung von Bildern mit natürlichen Szenen, wie z.B. Fotografien, jedoch weniger für Linienzeichnungen oder Cartoons. JPEG kann nur Einzelbilder hantieren, und zur Kodierung von Filmsequenzen wird das verwandte MPEG-Format verwendet. (Es gibt auch ein M-JPEG Verfahren für Bildfolgen, jedoch wird dieser nicht häufig verwendet).

Die JPEG-Komprimierung ist verlustbehaftet. Damit ist ein Bild im JPEG-Format nur eine Annäherung an das Original. JPEG nutzt physiologische Eigenschaften des menschlichen Auges bei der Komprimierung aus, indem beispielsweise kleine Farbunterschiede entfernt werden, da diese in geringerem Maße erkannt werden als kleine Helligkeitsunterschiede. Falls Bilder später mit einem rechnergestützten Verfahren analysiert werden sollen können diese kleinen Unterschiede zum Original Probleme in den Algorithmen verursachen.

Bei JPEG kann der Komprimierungsgrad angegeben werden, u.a. indem man den zulässigen Informationsverlust angibt. Bei der Bilderzeugung kann bestimmt werden welche Dateigröße und Bildqualität gewünscht wird. Bilder mit niedriger Qualität werden oft benutzt, um Bilder zu katalogisieren. Zusätzlich kann man bei der Dekodierung angeben, ob man Geschwindigkeit oder Qualität bevorzugt.

## 3.3 GIF

Das Bildformat GIF “Graphics Interchange Format” wurde von Comuserve zur hardwareunabhängigen Speicherung und Übertragung von Rasterbildern entworfen. Die Bilder werden in Einheiten und Untereinheiten in einem Datenstrom unterteilt, die die relevanten Daten und Parameter enthalten. Da die Einheiten im allgemeinen voneinander abhängig sind, sollte ein Enkoderprogramm Einheiten mit ähnlichen Parametern nahe beieinander ablegen. Bei GIF ist es nicht möglich Übertragungsfehler zu erkennen oder gar zu korrigieren. Obwohl GIF ursprünglich nicht dafür vorgesehen ist, können in gewissem Maße Animationen damit gespeichert werden.

# 4 Techniken auf der Browser-Seite

Wir haben bereits beschrieben, wie mit Hilfe von Plug-Ins und externen Anwendungen die Funktionalität eines Browsers erweitert werden kann. Im weiteren werden die Techniken vorgestellt um Interaktion und Logik in Web-Seiten zu beschreiben.

## 4.1 Java und Java-Applets

Java ist eine objektorientierte Programmiersprache, die eine Sonderstellung im WWW einnimmt. Ursprünglich wurde sie von SUN Microsystems unter dem Namen **Oak** eingeführt, um Programme für elektronische Apparate zu entwickeln. Mit Hilfe von Java können Interaktionen in Web-Seiten programmiert werden. Die Java-Programme werden zu einem sy-

stemunabhängigen Code übersetzt, der in einer Java Laufzeitumgebung abläuft, die für eine Vielzahl von Systemen und die meisten Browser vorhanden ist. Damit können Programme in Java im wesentlichen unabhängig von der Zielplattform entworfen werden. Java Applets können in Web-Seiten integriert werden, so daß diese von einem Web-Server in den Web-Client heruntergeladen und anschließend dort ausgeführt werden. Wichtige Kennzeichen sind:

- Java Applets sind ein Beispiel für mobile Programme, die nach Aufforderung auf den Zielrechner übertragen und ausgeführt zu werden.
- Java Applets machen Web-Applikationen zu echten Client/Server-Anwendungen, da Java-Code die entsprechende Logik auf dem Client ausführen kann.
- Zugang zu Datenbanken kann ohne ein vermittelndes Gateway-Programm durchgeführt werden, das auf einem Server installiert ist. Ein Java-Programm kann Anfragen direkt an eine Datenbank, beispielsweise mit JDBC senden.
- Animationen, Spreadsheets, geschäftliche Transaktionen, Grafiken, usw. können direkt im Browser ausgeführt werden.

Bei der Entwicklung verteilter Anwendungen in Java wird die Kommunikation oft durch eigene Funktionen implementiert, wenn ein Java Applet auf einem Browser eine Verbindung zu einem Programm auf einem anderen Rechner benötigt. Diese Möglichkeit ist für einfache Kommunikationsmuster zu erwägen. Die zweite Möglichkeit ist die Verwendung einer Zwischenschicht wie z.B. RMI, CORBA oder DCOM. Hier gibt es inzwischen gute Entwicklungswerkzeuge, die bei großer Funktionalität leicht zu benutzen sind.

In Java können grafische Objekte und Benutzeroberflächen programmiert werden. Eigene, plattformunabhängige Bibliotheken implementieren die Basisfunktionalität. Neben reiner 2D-Funktionalität gibt es auch Bibliotheken mit 3D-Funktionalität.

## 4.2 JavaScript

JavaScript ist eine standardisierte Skriptsprache, die in einer Web-Seite integriert und innerhalb des Browsers ausgeführt wird. JavaScript bietet die Möglichkeit mit dem Browser und Web-Seiten Informationen auszutauschen. JavaScript wird von den meisten Browsern unterstützt. Es gibt auch andere Skriptsprachen für denselben Zweck, wie z.B. Microsofts VBScript. Der Unterschied in der Praxis liegt darin, daß diese Skriptsprachen nur teilweise unterstützt werden. JavaScript wird zumeist für reine Interaktion (d.h. Benutzerschnittstelle) benutzt und nur zu einem geringen Teil für die Programmlogik.

Die grafischen Möglichkeiten in JavaScript sind sehr beschränkt. Der Funktionsumfang entspricht in etwa dem von HTML. Allerdings ist es möglich, den Inhalt dynamisch, d.h. als Animation zu gestalten.

## 4.3 ActiveX

ActiveX-Controls ist eine von Microsoft entwickelte Technologie, um Programme in Browsern ablaufen zu lassen. Wie Java kann ein ActiveX-Control auch in anderen Umgebungen als die des Browsers ausgeführt werden. Der Anwendungsbereich für ein ActiveX-Control



in einer Web-Seite entspricht dem eines Java-Applets. Ein ActiveX-Control ist eigentlich ein COM-Objekt, das bestimmten Regeln für seine Interaktion mit der Umgebung folgt. Ein ActiveX-Control hat direkten Zugang zu Rechner-Ressourcen und das Betriebssystem wie bei einem gewöhnlichen Programm. Dies ergibt eine große Flexibilität, jedoch bedeutet das auch leichteren Zugang zu einer Maschine für Dritte, die Schaden anrichten können. Durch die enge Integration mit den Plattformen von Microsoft stehen ActiveX-Controls in der Praxis nur für die Windows-Plattform zur Verfügung. Diese Technologie wird aufgrund der weiten Verbreitung der Technologie von Microsoft oft eingesetzt. In der Praxis können durch die enge Integration mit der Client-Plattform Unterschiede bei der Ausführung auftreten, abhängig auf welcher Plattform das ActiveX-Control ausgeführt wird.

#### 4.4 Broadway

In der Version X11R6.3 des **X Window Systems** mit dem Codename **Broadway** können mit Hilfe eines Plug-In Anwendungen für X in einem Browser ablaufen. Dabei können vom Browser aus X-Anwendungen gestartet werden, deren Fenster entweder auf dem Desktop oder aber innerhalb einer Web-Seite ablaufen. Die Anwendungen haben dieselben Kennzeichen wie normale Anwendungen unter X. Mit Hilfe dieser Technik können X-Anwendungen auf dem WWW ohne Portierungsaufwand durch alleiniges Umlenken des Display ablaufen. Voraussetzung ist jedoch daß ein X-Server auf dem Client-Rechner läuft.

Da die Übertragung von Daten im X-Protokoll sehr datenintensiv sein kann wird eine komprimierte Version namens Low-Bandwidth-X (LBX) oder X.fast verwendet, womit der Datenverkehr zwischen Server und Client komprimiert wird. Es existieren ähnliche Technologien für mobile Anwendungen, z.B. das System Tarantella, das ebenfalls eine Erweiterung des X-Protokolls mit starker Kompression verwendet.

#### 4.5 VRML

VRML (Virtual Reality Modelling Language) wird verwendet, um dreidimensionale Geometriemodelle zu visualisieren. In Bezug auf Grafik ist VRML in der Version 1.0 eine Untermenge des OpenInventor 3D-Dateiformates. Zusätzlich gibt es Erweiterungen zum Navigieren (WWW inline und WWW anchor) und um den Detaillierungsgrad des 3D-Modells zu steuern. In einem Browser werden VRML-Modelle oft mit Hilfe eines Plug-In realisiert.

Eine VRML-Datei besteht aus einem Knoten im Inventor-Dateiformat, der aus einer wahlfreien Anzahl von Unterknoten besteht. Eine 3D-Szene besteht unter anderem aus Punktlichtquellen, Spots, gerichteten Lichtquellen, Kameras und Koordinatentransformationen. Als Geometrieprimitive gibt es Kugeln, Zylinder, Kegel und Polygon-Mesh, mit ihren Materialbeschreibungen und Texturen.

VRML 2.0 ist eine Weiterentwicklung, die zusätzlich eine interaktive Steuerung der Szene erlaubt. Dieser Standard bietet die Möglichkeit zu dynamischen und skalierbaren 3D-Umgebungen mit einem hohen Anteil von Objekten, die sich bewegen. Abstandsabhängiges Audio, verhaltensgesteuerte Modelle, Animationen und Sensoren gehören zu den neuen

Möglichkeiten, die den Cyberspace realistischer machen sollen. Die Programmierungssprachen Java und JavaScript können als Erweiterungen innerhalb von Szenen benutzt werden.

#### 4.6 CandleWeb

Eine Technik, die sich besonders zur Präsentation von Computergrafik auf dem WWW eignet, wurde von der norwegischen Firma CandleWeb (<http://www.candleweb.no>) entwickelt. Als Ausgangspunkt wurde die Programmiersprache Å verwendet, die im wesentlichen eine Untermenge von C ohne Strukturen und Zeiger darstellt. Dafür gibt es Erweiterungen für grafische Objekte, Polygone, Polysplines, Text, Bitmaps und Input-Objekte. Grafik und Variablen in solchen Programmen werden durch sogenannte empowered variables gekoppelt. Werteänderungen im Programm bewirken eine Änderung der damit verbundenen grafischen Objekte. Durch Übersetzung nach Java laufen Å-Programme direkt im Browser ab.

Es gibt auch speziell kodierte grafische Objekte mit dem Namen QDV (Quick and Dirty Vector Graphics), womit verhältnismäßig kleine Dateigrößen erreicht werden können, was die Transfer- und die Präsentationszeit reduziert. Die Präsentation von QDV in einem Browser geschieht durch ein Java-Applet, wobei GIF- und JPEG-Bilder, Linien, Polygone, Splines, Verweise, HTML-Text und Animationen verwendet werden können.

Es gibt ähnliche Techniken von anderen Anbietern, z.B. Macromedia Shockwave. Dort können laufen grafische Präsentationen in einem Plug-In ab.

#### 4.7 Multimedia

Die Standards innerhalb des Bereiches Multimedia behandeln vor allem die Beschreibung, Kodierung und Synchronisierung von Medienströmen, und haben daher nur bedingt eine Bedeutung für unser Thema. Es gibt jedoch mehrere Vorschläge, Multimedia und Digitales Fernsehen mit dem WWW zu verbinden. Die Standardisierungsvorschläge von MPEG (Motion Picture Expert Group) (insbesondere MPEG-4 und MPEG-7) und MHEG (Multimedia and Hypermedia Information Coding Expert Group) umfassen Multimedia von der Codierung über Synchronisierung bis zur Präsentation des Inhalts.

SMIL baut auf HTML auf und beschreibt den logischen Aufbau von Präsentationen mit mehreren Medientypen und die Synchronisierung von Datenströmen. Andere Vorschläge, die auf HTML aufbauen gibt es für den Bereich des digitalen Fernsehens, wobei versucht wird WWW und TV-Technik zu integrieren (z.B. ATVEF), und eine gemeinsame Benutzeroberfläche zu entwickeln.

### 5 Serverseite

Anwendungen, die über die Präsentation von bereitgestellten Daten hinausgehen, benötigen eine Verarbeitung von Daten auf der Serverseite. Prozesse auf der Serverseite haben zur Aufgabe Anfragen entgegenzunehmen und das Ergebnis dieser Anfrage an den Client zu

übermitteln. Wir unterscheiden zwischen HTTP-Anfragen, die durch CGI bearbeitet werden, Standard-Middlewarelösungen wie CORBA und MPEG, sowie proprietäre Lösungen wie z.B. RealVideo.

## 5.1 HTTP-Anfragen

Eine komplexe Anfrage wird in der Regel mittels CGI durchgeführt. Sie wird an einen Web-Server geschickt, wo die Daten mittels CGI an einen Prozess weitervermittelt werden. Das Ergebnis wird berechnet und anschließend an den Browser geschickt. Der dabei ablaufende Prozess kann in einer beliebigen Programmiersprache programmiert sein. Die Details der Abarbeitung variieren jedoch von Installation zu Installation.

Bei der einfachsten Lösung wird ein eigener Prozess gestartet, der die Berechnung durchführt. Andere Lösungen haben ein Run-Time-System, das einen Pool von verfügbaren Prozessen bereithält, um den Durchsatz zu erhöhen. Diese Methode wird bei Java-Servlets verwendet. Microsofts ASP (Active Server Pages) verwenden dagegen spezielle Skript-Programme, die im Web-Server installiert werden.

Ein Prozess, der durch CGI gestartet wird, bekommt client-, server- und anfrage-spezifische Kontextinformationen über Umgebungsvariablen. Diese Variablen geben Auskunft über die verwendeten Programme, den Benutzer, Datentypen, Namen von Ressourcen, etc. Mit Hilfe der Post- oder der Get-Methode in HTTP und Forms in HTML können auch anwendungsorientierte Variablen übertragen werden. Variable können auch als Teil eines URL übertragen werden. Mehr zum Thema CGI findet man in der Vielzahl praktisch orientierter Handbücher, z.B [Wei96].

Eine einfache, weniger flexible Lösung stellen die *server-side includes* (SSI) dar. Diese sind spezielle Markierungen in einer HTML-Datei, die durch den aktuellen Inhalt ausgetauscht werden, bevor sie zum Browser geschickt werden. Sie werden normalerweise benutzt, um Dateinamen, Datum, etc. in eine Dokument einzufügen, eignen sich aber auch für Überschriften, Menus oder andere grafische Elemente.

## 5.2 Middleware und standardisierte Protokolle

Verschiedene Middleware-Lösungen werden verwendet, um Daten zwischen Server und Client zu schicken. Diese werden oft als eigene Anwendungen programmiert, jedoch können sie auch Teil des Servers sein. Die Kommunikation von und zu den aktiven Teilen der Anwendung wird nicht direkt durch HTTP geführt (obwohl HTTP als Träger benutzt wird). Oft werden Inter-Prozess-Aufrufe wie in CORBA, RPC oder DCOM, sowie Stromdaten wie in MPEG verwendet. Dies hat den Vorteil, daß eingebaute Mechanismen für die Kommunikation verwendet werden können, die in HTTP fehlen.

## 5.3 Proprietäre Lösungen

Diese werden normalerweise für spezielle Anwendungsgebiete eingesetzt. Als Beispiel führen wir RealVideo an, wobei Audio- und Video-Daten über das Internet geliefert werden, und das WWW als Benutzerschnittstelle dient. Ebenso bieten einige Web-basierten

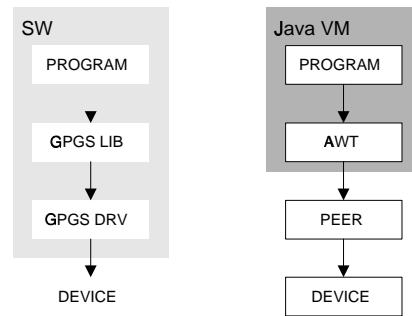


Abbildung 2: GPGS-Modell und Applet-Modell im Vergleich

Entwicklungswerkzeuge eigene Server an, die Daten zu und von speziellen Clienten-Programmen (Plug-Ins) transportieren.

## 6 Integration von Grafikbibliotheken mit dem WWW

In diesem Abschnitt untersuchen wir, wie die Grafikbibliothek GPGS und das WWW vereinigt werden kann. GPGS steht in diesem Zusammenhang als Beispiel für eine Grafikbibliothek mit einem allgemeinen Funktionsumfang.

GPGS kann in vier Teile untergliedert werden: das Anwendungsprogramm, die GPGS-Bibliothek, Device-Treiber und das Device (siehe Abb. 2, linke Seite). Innerhalb eines Applet in Java kann das Grafiksystem ebenfalls in vier Hauptelemente eingeteilt werden: Das Programm, AWT-Komponenten mit den entsprechenden Windows-Komponenten (peers) und das (siehe Abb. 2, rechte Seite).

Ein wesentlicher Unterschied zwischen GPGS und dem Applet-Modell ist der Zugang zum Bildschirm. Eine allgemeine GPGS-Anwendung hat normalerweise Zugang zum gesamten Bildschirmbereich, während ein Java-Applet nur bestimmte Bereiche auf dem Schirm adressieren kann. So kann ein Java-Applet zwar eine Web-Seite im Browser anzeigen lassen, jedoch liegt diese Seite außerhalb des Zugriffs des Applets selbst.

Falls ein Java-Applet Aufrufe der GPGS-Bibliothek benötigt, muß auf die Integration zwischen den Aufrufen des Java-Teils (im wesentlichen AWT-Komponenten) und den von GPGS erzeugten Aufrufen geachtet werden. Im wesentlichen gibt es zwei Möglichkeiten zur Integration zwischen dem Applet-Modell und dem GPGS-Modell.

1. Das Applet ruft die GPGS-Bibliothek auf. Die Daten werden dort bearbeitet und das Ergebnis wird über das Applet an die AWT weitergeleitet
2. Das Applet ruft die GPGS-Bibliothek auf, wo die Dateien innerhalb des GPGS-Modells bearbeitet werden.

Die beiden Möglichkeiten unterscheiden sich im Grad der Integration mit dem Fenstersystem, also in welchem Grad das Java-Programm Zugang mittels AWT und nicht nur durch

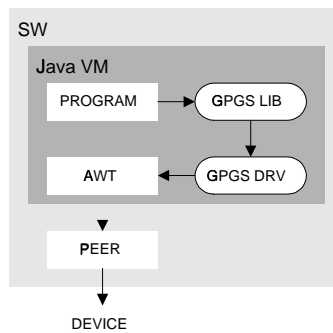


Abbildung 3: Verwendung von nur Java

GPGS erhält. Die erste Variante läßt keine Integration mit dem Fenstersystem zu. In der zweiten Variante kann für Teile eines Programms AWT benutzt werden, während andere Teile die GPGS-Funktionen benutzen.

## 6.1 Reine Java-Lösung

Die erste Möglichkeit zur Integration zwischen GPGS und Java ist die Implementierung der gesamten Bibliothek in Java, inklusive der Erstellung der Treiber, die GPGS an AWT koppeln.<sup>3</sup> Siehe dazu Abb. 3. Eine solche Lösung kommt mit einer reinen Java-Laufzeitumgebung aus und gibt eine gute Fenstersystemintegration diese Programme. Jedoch ist der Portierungsaufwand sehr hoch. Weiterhin ist es fraglich, ob die bestehenden Java-Implementierungen sich für größere grafische Anwendungen eignen.

## 6.2 Lokale Bibliothek

Eine zweite Möglichkeit ist die Definition einer JNI-Schnittstelle zwischen der bestehenden GPGS-Implementierung und Java. Das Programm wird dabei in Java implementiert, während alle Aufrufe von GPGS außerhalb der virtuellen Maschine des Client ablaufen.

Ohne Fenstersystemintegration wird ein JNI-Interface definiert, das Aufrufe vom Java-Programm an GPGS erlaubt. Eine Lösung mit Fenstersystemintegration fordert zusätzlich, daß ein eigener GPGS-Treiber und ein Java-Stub entwickelt werden, die das Ergebnis im Applet-Modell weitersendet. Bestehende Bibliotheken können damit unverändert weiterverwendet werden, müssen jedoch zusätzlich auf allen Clients einzeln installiert werden.

<sup>3</sup>Die konzeptuelle Unterteilung zwischen Bibliothek und Treiber wurde beibehalten, obwohl AWT Funktionen besitzt, die die Aufgaben des GPGS-Treibers übernehmen könnten.

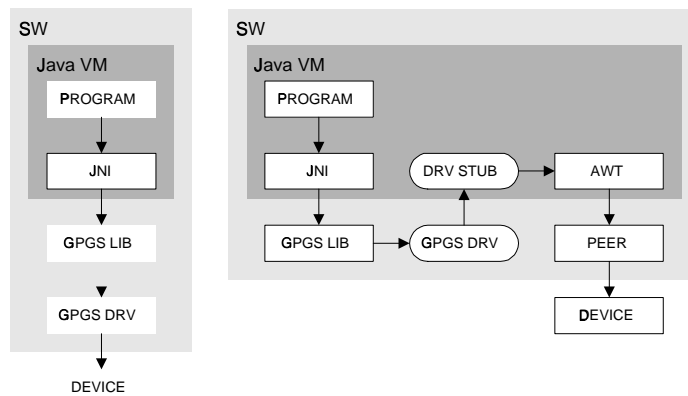


Abbildung 4: Lokale Bibliothek ohne (links) und mit (rechts) Fenstersystemintegration.

### 6.3 GPGS-Server

Eine dritte Möglichkeit besteht in der Errichtung eines eigenen Servers für GPGS-Anfragen. Dabei kann ein ORB (Object Request Broker) aus einer CORBA-Implementierung oder eine andere Form der Kommunikation zwischen Java-Client und GPGS-Server verwendet werden. In der Implementierung erzeugt man einen Stub im Client, der alle Aufrufe an GPGS abfängt und an den GPGS-Server weiterleitet. Dort wird die Anfrage von einem weiteren Stub aufgefangen und an die behandelnde Einheit weitergeleitet. Das Ergebnis wird an einen Treiber-Stub geschickt, der es an den Client für die weitere Bearbeitung weiterleitet. Die Methode kann ohne (Abb. 5) oder mit (Abb. 6) Fenstersystemintegration implementiert werden.

Mit einer solchen Lösung können bestehende GPGS-Implementierungen weiterhin benutzt werden. Der GPGS-Server kann auf einem Rechner installiert sein, der für grafische Anwendungen geeignet ist, und mehrere Clients können einen einzigen Server benutzen. Jedoch muß jeder GPGS-Aufruf durch einen ORB (und damit wahrscheinlich über ein Netzwerk) geschickt werden. Die Nachteile können gemildert werden, indem Optimisierungen in Form von Caching oder Umgruppierung vorgenommen werden.

In diesem Fall ist die Lösung mit Fenstersystemintegration besonders interessant, da Teile des Graphiksystems damit lokal im Client wie in einem gewöhnlichen Applet ablaufen kann. Damit sind nur Aufrufe an GPGS von einem verteilten ORB abhängig.

### 6.4 Alternative Fenstersystemintegration

In den bisherigen Lösungen ist die Fenstersystemintegration dadurch realisiert daß die Ergebnisse von GPGS in das Applet-Modell zurückgeführt wurden. Eine Alternative dazu ist AWT durch die entsprechenden GPGS-Funktionen zu ersetzen (siehe Abb. 7). Jedoch ist es dabei nicht zulässig Klassen in Paketen zu benutzen, die mit "java." beginnen (x.B. "ja-

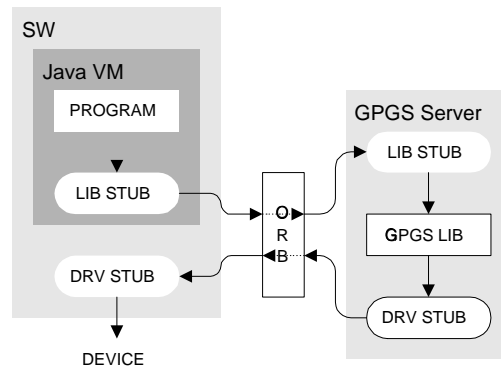


Abbildung 5: GPGS-Server ohne Fenstersystemintegration

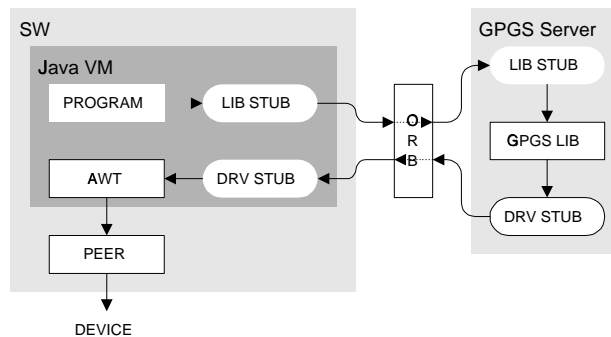


Abbildung 6: GPGS-Server mit Fenstersystemintegration

va.awt"). Diese Lösung kann dennoch benutzt werden, indem beispielsweise alle entsprechenden Klassen abgeändert werden (java.awt ersetzen, Referenzen durch entsprechende AWTX-Referenzen ersetzen, oder Installation eines speziellen java.awt im Client).

## 6.5 Bestehende Anwendungen

Um bestehende Anwendungen im Web zugreifbar zu machen wird die Eingabe des Benutzers in einem Browser aufgefangen und zur Anwendung geschickt. Nach der Berechnung wird das Ergebnis dort aufgefangen und an den Browser vermittelt.

Wir nehmen uns zunächst bestehende Anwendungen mit einer einfachen statischen Struktur vor. Die Lösung paßt am besten für Anwendungen, die in einer Art Stapelverarbeitung vorgehen: Zuerst wird der Großteil der Interaktion mit dem Benutzer durchgeführt, während die Bearbeitung auf einem Server und die Präsentation auf dem Client nachfolgt.

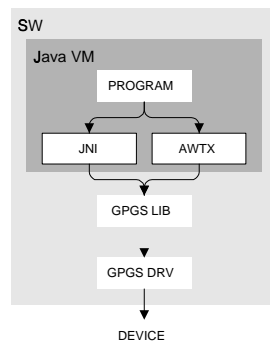


Abbildung 7: Alternative Fenstersystemintegration

Komplexere Interaktionsmechanismen benötigen im Allgemeinen den (Programm-)Zugriff auf die innere Struktur der bestehenden Anwendungen.

**Statische Programmstruktur.** Hier sind die Anzahl der Eingabeparameter und das Aussehen des Resultats fest bestimmt. Die Parameter werden in einem Client aufgefangen und an die Anwendung geschickt. Nachdem das Ergebnis bereitsteht, wird dieses aufgefangen und an den Client übertragen.

Einfache Parametereingaben können in HTML-forms abgewickelt werden, während Java oder andere Programmieretechniken im Client für komplexere Fälle benötigt werden, um Abhängigkeiten oder Validierungsregeln zu überprüfen, bevor die Parameter zum Empfänger auf dem Server gesendet werden. Alle Parameter aus der Anfrage werden gesammelt zur Anwendung übertragen. Je nach Anwendung können Konfigurationsparameter, ein API oder Screen-Scraping benutzt werden. Die ersten beiden Möglichkeiten werden am ehesten für Programme, die GPGS benutzen in Frage kommen.

**Konfigurationsparameter** werden zumeist in Batch-orientierten Programmen benutzt. Die Parameter werden bei Programmstart übertragen. Dies können reine Startparameter (Kommandozeile), Konfigurationsparameter in einer Umgebungsvariablen, oder auch ein Teil einer Pipe (Interprozesskommunikation) sein.

**API** sollten für bestimmte Bibliotheksaufrufe, ORB-Aufrufe oder für lineare Protokolle (über ein Netzwerk hinweg) benutzt werden, falls solche vorhanden sind.

**Screen-scraping** bedeutet den Bildspeicher oder eine andere Repräsentation des Schirmbildes abzulesen. Diese Methode hat ihre Berechtigung, wenn alle anderen Möglichkeiten versagen. Dabei werden Parameter durch die Mechanismen der Anwendung eingegeben. Die Methode fordert eine hohe technische Integration gegenüber der Infrastruktur, die von der Anwendung benutzt wird. Jedoch ist sie sehr effektiv, falls diese Infrastruktur wohlbekannt ist.



Nach der Eingabe wird die Anwendung wie üblich fortgesetzt, abgesehen davon, daß das Resultat aufgefangen werden muß. Dies kann durch eigene Treiber, APIs oder durch Screen-Scraping durchgeführt werden. Für Anwendungen, die GPGS benutzen, könnte der vorhandene JPEG-Treiber eine angemessene Lösung sein.

**Eigene Treiber** können Ergebnisse an den Client senden, entweder direkt (an ein Plug-In, o.ä.) oder als ein Rasterbild, das in eine Web-Seite integriert wird.

**API** können wie Schirmtreiber benutzt werden. Wie zuvor werden die Ergebnisse aufgefangen und auf geeignete Weise formatiert bevor sie zum Client gesendet werden.

**Screen-scraping** kann auch in diesem Fall benutzt werden, um das Ergebnis vom Schirm abzulesen. Das Ergebnis wird auf geeignete Weise formatiert (beispielsweise als ein Rasterbild) und an den Client geschickt. Dies setzt allerdings den Zugang zu diesen Ressourcen voraus.

Diese Lösungen haben gemeinsam, daß sie zusätzlich zum Anzeigen eines statischen Resultats die Interaktionsmechanismen des Client verwenden können. Beispiele sind die Wahl von Ausschnitten, oder der Gebrauch von ebenfalls erzeugten Image-Maps, um weitere Eingaben vom Benutzer entgegennehmen zu können.

**Dynamische Programmstruktur.** Von der dynamischen Struktur einer Anwendung spricht man, falls die Eingabeparameter im Voraus nicht bestimmt werden können. Dies kann bei extrem großen Parametermengen der Fall sein, wenn nur begrenzte Teile in einem Durchgang benötigt werden, oder wenn sich die Eingabeparameter oft ändern. In solchen Anwendungen muß das Ergebnis zur Definition der Benutzerinteraktion interpretiert werden. Daher erzeugt man eine Schleife, in der die Anwendung ein Schirmbild zur Parametereingabe erzeugt, das aufgefangen und interpretiert wird. Darauf beruht die Erzeugung des Bildes für den Client. Die Eingabeparameter werden aufgefangen und an das Programm zurückgeschickt. Damit wird ein neuer Durchgang gestartet. Siehe dazu Abb. 8.

## 7 Ausblick

Grafische Methoden werden in vielen Anwendungen eingesetzt, die eine grafische Benutzeroberfläche besitzen. Nach und nach werden diese Anwendungen mit unterschiedlichen Techniken für das WWW zugänglich gemacht. Für ältere Grafikbibliotheken, die bisher keine Unterstützung für das WWW hatten, gibt es mehrere Möglichkeiten um Anwendungen, die solche Bibliotheken benutzen, in das WWW zu integrieren. Die Wahl der Methode ist jedoch von der Anwendung und anderen Anforderungen abhängig, so daß keine allgemeingültige Empfehlung gegeben werden kann.

Das WWW wird sich weiter entwickeln, und eine neue Herausforderung zeichnet sich ab, indem in größerem Maße mobile Terminale wie PDA oder Mobiltelefone verwendet werden. Die Möglichkeiten für die Benutzung von Computergrafik werden hier rasch wachsen. Dazu werden Proxy-Lösungen und Konverter notwendig sein, um die relativ niedrige Bandbreite und Begrenzungen in den grafischen Fähigkeiten dieser Geräte besser auszunutzen.

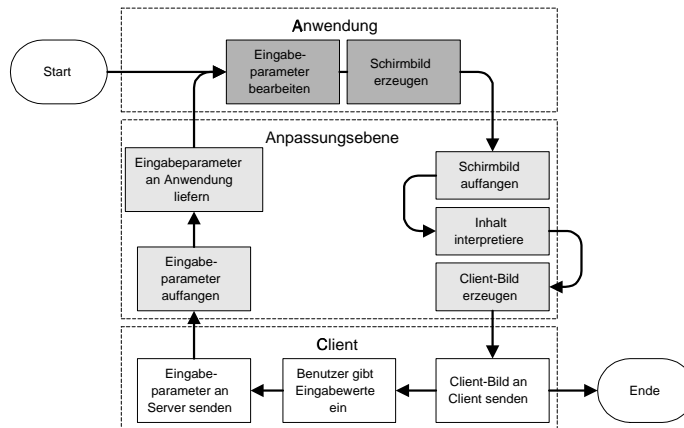


Abbildung 8: Ausführung bei dynamischer Programmstruktur

Solche Geräte sind darauf hin optimalisiert klein und mobil zu sein, anstatt Grafik mit hoher Qualität zu präsentieren. Die Grundlagen der hier vorgestellten Techniken ermöglichen einen nahtlosen Übergang auf mobile Endgeräte, damit Anwendungen gleichermaßen auf Bürorechnern und auf mobilen Terminalen zugänglich sind. Hier wird weiterhin Forschung und das Ausprobieren neuer Methoden benötigt, um den nahtlosen Zugang auf Dokumente und andere Ressourcen zu ermöglichen.

Die Untersuchung "Grafik auf dem Web" ist ein erster Schritt, Techniken zur Nahtlosigkeit für bestimmte Anwendungen vorzustellen. Indem als einzige Benutzerschnittstelle das WWW benutzt wird, ist bereits der erste Integrationsschritt erreicht. Für die Zukunft wird es dann nur noch notwendig sein, die WWW-Schnittstelle auf neue technische Entwicklungen anzupassen.

## Literatur

- [BL96] T. Berners-Lee. *Axioms of Web Architecture: 1.* Web Document, <http://www.w3.org/DesignIssues/Axioms.html>, 1996.
- [BLG97] T. Berners-Lee und B. Girschweiler. *Graphics formats for the World Wide Web.* Web Document, <http://www.w3.org/Graphics/>, 1995, 1997.
- [Net98] Netscape. *The LiveConnect/Plug-in Developer's Guide.* web pages, <http://home.netscape.com/eng/mozilla/3.0/handbook/plugins/index.html>, 1998.
- [Uma97] A. Umar. *Object-Oriented Client/Server Internet Environments.* Prentice Hall, Upper Saddle River, NJ, 1997.
- [Wei96] W. Weinmann. *The CGI Book.* New Riders Publishing, Indianapolis, 1996.