Abstract

World Wide Web Hypermedia pages contain text, still images and dialog elements. We present a method to integrate video and animation into WWW pages without any modifications to existing standards or protocols. This scheme is based on the multi-image capabilities of image coding standards. Together with the standardized dialog elements of HTML the system provides interactive video in the World Wide Web.

We present an application of this technology for remote control of mechanical equipment. As an example we enable every Web-user to watch and operate a real model railroad in our University lab. Another application taking advantage of realtime animated computer graphics and video is remote visualization.

I. Relevant Web Basics

I.A. The World Wide Web

The World Wide Web (WWW) [13] is a global, distributed hypermedia system. Hyperdocuments are non-linear documents; they contain links to other documents. Documents of the WWW are not only restricted to text. Other media such as images, sounds, and movies can be referenced. The layout of a page is controlled by an HTML-type document. HTML, the Hypertext Markup Language [5] is a text based document description, defined by an SGML document type definition (DTD) [3].

The Web's functional model is the client-server paradigm. Documents are exchanged between WWW-clients and WWW-servers using the Hypertext Transfer Protocol (HTTP) [1]. The currently used protocol version HTTP 1.0 relies on a stateless request-response mechanism. A transport system connection is established for the transfer of every document. The client requests the document by its name and the server responds with the document's data. The interactive elements such as forms use the same request-response mechanism. There is no continuous connection between client and server.

The World Wide Web is rapidly expanding. It is already the largest hypertext system world-wide. Such extensive usage demands more features and flexibility than the system provides today. The system has been designed as a distributed information system which links locally available documents into a global hypertext document. But many applications, especially in the commercial domain require enhanced features like security, interactivity, transport efficiency, improved layout capabilities, and control over the client display. Some of these features, e.g. interactive elements and security have already been added to the World Wide Web. Others (HTTP 2.0) are in the design phase or under discussion. Most enhancements are based on extensions to the used standards or even replacements for them.

I.B. Media in the WWW

WWW mechanisms are media independent. The HTTP protocol is transparent for the type of documents. Client and server negotiate media types using the MIME (Multipurpose Internet Mail Extensions) model. An HTTP request accesses documents independent from the document type only by name. The server decides about the document type and returns a MIME type specification with the HTTP response. It is the client's responsibility to select the proper presentation method according to the document type.

WWW pages currently contain text, graphics and dialog elements. Presentation of other data and media is accomplished by external programs, so-called viewers. WWW clients do not directly support synchronous playback of stream oriented media such as audio and video. Audio and Video are treated as files. They are processed in three phases: retrieve, store in the file system, and presented through a viewer. Usually, the presentation of such data begins after the document has been retrieved completely from a server. Thus continuous media cannot be put directly (inline) on a WWW-page.

Applications which require continuous updates of the clients view force the user to repeatedly retrieve a one-shot document, that is constantly updated in the server site [2]. Discussion on the inclusion of inline media other than graphics, e.g. audio [12] and video, is currently quite intense. How this integration can be achieved is subject to active research. New protocols, protocol elements and extensions to HTML have been proposed [11],[7]. However, changes to existing and widely used standards such as the WWW protocols and formats should be done very carefully. If possible, they should be avoided altogether.

I.C. Image and Movie Formats

Although the WWW supports many media formats only a small number of formats is actually used. By far the most widely used image format is Compuserve's GIF (Graphics Interchange Format [4]). GIF has been designed for the Compuserve network and has been adopted by the internet community. The internet does not exploit all capabilities of the standard. It uses only a subset of the features required to encode images. The next chapter will explain this topic in detail. Other commonly used
image formats are JFIF (JPEG File Interchange Format) of the Joint Photographic Experts Group [6] and the X Window system's text based bitmap encoding (XBM). The JPEG encoding scheme is superior to GIF if applied to continuous-tone images. Its usage is growing fast since major WWW clients (browsers) support inline JPEG images.

The usual movie format in the Web is the ISO standard MPEG (Moving Pictures Expert Group) [8]. Quicktime and Video for Windows are other movie formats in use, but they are not vendor and platform independent. Nevertheless they can be decoded on nearly all platforms. This depends on the availability of the proper decoder software on each client system. The decoders are external programs because these data types are not decoded directly by WWW clients, resulting in offline presentation of the movie after it is downloaded. This mechanism to support moving pictures is therefore not adequate for the transmission of live video and interactive video applications.

**II. Pushing the Limits of Standard Mechanisms**

**II.A. Moving Images**

All components of the World Wide Web are currently developing. One of the directions is integration of stream oriented media into WWW pages. Some movie delivery projects in the WWW propose inline decoding of MPEG. Most of them use pre-compressed MPEG sequences. Remote control and remote visualization are applications which require either realtime generated animated computer graphics unsuitable for MPEG compression or the generation of live video without the use of expensive MPEG compression hardware.

Moreover MPEG decoders are not readily available in the installed base of personal computers connected to the internet. In contrast to MPEG, still image decoders (JPEG, GIF) are readily available in the WWW clients. They can be used to show sequences of images. Such sequences appear to be videos if the images replace each other at a reasonable rate.

Regardless of the actual chosen video stream format a client has to support continuous - live - decoding explicitly. The WWW client has to be able to retrieve documents concurrently and present them continuously as the data arrives. The architecture of the WWW clients must be event and data driven. It must not block while retrieving a document.

**II.B. Sequence of Still Images**

One method for animated graphics is the consecutive transmission of individual images. Instead of terminating the transport system connection after transmission of one image, the server continues to transmit images until the client terminates the connection. Not only images can be animated this way, but also other document types, e.g. text. This leads to a new architecture for the HTTP connection where more than just one document is transferred during the lifetime of the connection. However this animation method has some disadvantages.

Combination of a series of documents to a single structure which fits into an HTTP response requires a description of the contents of the response. The result is a new layer of control structures between the document layer and the HTTP protocol. Existing WWW clients and servers have to be modified in order to deal with streams of documents instead of single documents.

The method allows to exploit the existing decoder software of WWW clients. It works with all image formats including GIF and JPEG. A major disadvantage of such a pseudo animation, if used for moving images, is the fact that always entire images have to be encoded and transmitted. The encoded animation contains considerable redundancy. All images are encoded completely independent from each other. They all contain header information and often encode the same unchanged image parts over and over again.

**II.C. Image Streams as a solution**

Some image formats are not only restricted to single images. They describe the encoding of a sequence of consecutive images. Examples are the PDS and GIF formats. The GIF format has been designed to serve for animated graphics in the Compuserve network. Recent specifications allow for an infinite sequence of images. The images can be of different size and depth within a global rectangle. The possibility to encode differently sized image parts permits simple frame differencing as optimization method. Of course, the extent of data reduction due to frame differencing depends on the encoded sequence. But a large number of applications especially in the remote control and process visualization area rely on animated graphics displays with only slight changes from image to image. Many publicly available GIF-viewer support this feature. It is indeed possible to present a movie with these viewers.

An image stream encoded in a multi-image format is accessed via hyper links as any other document. The WWW client opens the transport system connection, sends the HTTP request and waits for the response. The response contains HTTP header information and the image stream as HTTP body. The HTTP header indicates the document type to the client. The document type is encoded in a string consisting of type and subtype, e.g. image/gif. This document type description does not have to be changed in order to support multiple images. A GIF encoded image sequence has the same document type as single images. A single image is regarded as special case of a sequence containing only one image.

The main advantage of image stream formats compared to sequences of separated images is the possibility to exploit format specific optimization methods like frame differencing. In addition image streams are backward compatible to WWW clients.
which do not support moving images. Clients, which do not support the multiple image feature of multi-image capable formats but tolerate them, will show the first image of the sequence. They will terminate the connection after having decoded the first image or just stop decoding.

III. Implementing WebVideo

III.A. Client Software

WWW clients supporting the multi-image option of multi-image graphics formats will automatically present a video when decoding an image sequence. This means for the most widely used image format, GIF, that clients have to support at least the GIF87a specification. Most WWW clients do not support this specification entirely, i.e. they ignore the multi-image feature. This is due to the fact that GIF has been used in the internet only for the encoding of still images.

![GIF Image Format](image)

**Figure 1:** The format of a GIF image (upper) and a GIF stream (lower).

However, treatment of multiple images can easily be added to existing GIF decoder software. A GIF sequence consists of a global header and a series of separately encoded images (Figure 1). The difference between a GIF stream and a GIF image is only the number of images following the global header. There is no additional information or control structure accompanying the existence of more than one image. The global header does not contain information about the number of subsequent images.

An investigation of the available GIF image decoder software showed that the required changes for an upgrade from an image decoder to a stream decoder are minimal. We modified the GIF decoders of two publicly available WWW clients (Chimera, Mosaic) in such a way that they continue decoding as long as the GIF data stream does not terminate. The modified versions present moving images on a WWW page as long as the WWW server continues to send images.

III.B. Server Software

A video stream is accessed like any other document via its universal resource identifier or locator (URI/URL). The server transmits documents regardless of type and size. It does not notice if a document is written to disk or directly decoded by the client. A video stream will be transmitted like any other document. For the simplest case of a pre-recorded video which has been encoded in a multi-image capable format there are no changes necessary to the server system.

Usually WWW servers try to transmit documents as fast as possible. For that reason the frame rate at the client display depends on the quality of the transport system connection. The rate may be too high in a local environment and too low over slow links. The first case results in time-compressed presentation of the video. The later creates a backlog of frames, defeating the realtime capabilities. Our experiments showed that some changes to the server system are very useful in order provide controlled delivery of image sequences.

Different approaches are possible to show image sequences. The first is the so-called 'best effort' method. Best effort means transmission and decoding as fast as possible with the assumption that the connection is either not or just fast enough for realtime display. If a transport system connection is not fast enough for realtime display then the frame rate will be too slow. On the other hand such a system will show each image of a sequence. It will not skip any frame. The second method is time synchronisation. A playback system with time controlled presentation tries to show only the images of a sequence which fit into the time scale of the video. It will skip images if transmission, decompression, or display are too slow and it will delay playback in the opposite case. A combination of both approaches is best effort with upper limit, which tries to reach realtime display and limits the frame rate to an upper bound.

All methods except 'best effort' require a component which controls playback at the client. In addition this component needs a feedback mechanism between client and server in order to avoid overrunning of the client's buffering capabilities. Synchronization by the client would require a major change to the clients' software.

We therefore propose frame-rate control by the server. A software module in the server controls the transmission speed between video source and the WWW client. The flow control mechanisms of TCP/IP can be exploited for the feedback from client to server.

III.C. Server Extension for Framerate Control

The World Wide Web software provides a standardized method to include server side extensions.
This interface is the so called Common Gateway Interface (CGI). It has been designed to allow access to other information systems than WWW, e.g. WAIS. But the CGI can be used to feed all kind of data into the WWW system. In principle the CGI is an extension of the WWW name space of Universal Resource Identifiers to cover not only files but also the output of executable programs.

We designed and implemented a CGI program as a video extension to the server. This rate control CGI program feeds video streams to HTTP servers. It is invoked by the server in response to a request for an image stream from the client. The CGI program provides high flexibility allowing different synchronisation mechanisms, video sources and output formats.

![Figure 2: HTTP servers extensions through the common gateway interface (CGI). The URI points to an executable program (right) instead of a file.](image)

Supported rate control mechanisms are:
- best effort, (chapter 3.2)
- best effort with rate limit,
- near realtime presentation using frame skipping or delayed transmission.

Rate control mechanism, frame rate and upper bounds are adjustable by the provider of the hypertext documents. They can either be encoded into the URI or be fixed to a certain value. Encoding into the URI allows flexible adjustment for different streams.

Supported output formats are:
- a graphics stream encoded in GIF (chapter II.C.),
- sequences of still images encoded in JPEG or GIF (chapter II.B.),

The encoding for sequences of still images follows the specification of Netscape for server-push animation [9]. The default output format is GIF. However the CGI program automatically recognizes if the connected client supports the JPEG format. In this case the still images contained in the sequence will be encoded in JPEG rather than GIF due to the smaller data size.

Our stream server supports several video sources:
- a file,
- a portion of a file addressed by frame number,
- a stream from a system queue,
- a single image out of an image stream from a system queue.

Besides rate controlled playback of pre-recorded sequences the video extension is able to retrieve an image sequence from a system queue. The system queue is a named shared memory space which can be

![Figure 2: HTTP servers extensions through the common gateway interface (CGI). The URI points to an executable program (right) instead of a file.](image)
A video stream from a live source has to be encoded in one of the supported image or stream formats. The encoding is performed only once per stream. Each encoded image of a stream is put into a shared memory space where it is accessible for the synchronizing CGI programs. Many instances of stream synchronizers retrieve the encoded images simultaneously from the shared space of the image queue. They may even retrieve different images at the same time to keep up with the state of their connections. Encoding the stream only once allows many clients to connect to a live source at the same time without overloading the server.

An image stream from a live source is encoded by a so-called stream server. The stream server provides the shared memory queue to the synchronizing CGI programs. It connects at the front-end directly to the live video source and on the back-end to a number of stream synchronizers via the shared memory queue describe above. The main purpose of the stream server is conversion from the image data format of the video source to the desired stream format for transmission to the WWW clients. The front end consists of image decoder modules which accept different image formats. The back end is currently equipped with GIF and JPEG encoders.

IV. An Interactive Application

Live images are often produced by monitoring systems. Either video recorded by a camera or computer graphics generated by software. Many remote monitoring systems allow interaction of a user with the monitored system. Such remote control requires a command channel from the display to the monitored system. The command channel carries user input to the system. The controlled system's feedback is visually given as a stream of computer graphics or live video from a camera.

IV.A. User Interaction

The World Wide Web supports user interaction via forms. Forms are encoded as part of the base HTML document. They can contain different user interface elements like buttons, menus and text input fields. User input is not transmitted continuously. It
is collected by the client and transferred to the server on request of the user. The transmission of user input fits into the request-response mechanism of HTTP. The collected results of user interaction are encoded in a URI and sent with the HTTP request. The server extracts information from the URI, processes it using a server extension and usually returns the HTTP response. The response carries a confirmation for the user input as a new WWW page which replaces the old one.

![Diagram](image)

**Figure 5:** User commands evaluated by server extensions control system parameters. Visualization of feedback fed into the WWW by another server extension.

Explicit command confirmations are not necessary if feedback is given visually as a graphics stream. A remote control WWW page will contain animated graphics showing the state of the remote system and forms for user input. The page will not be changed to show command confirmation. Instead the WWW server is forced by its forms evaluating extension to return an empty HTTP response (using the 'No Response' response code 204). The client will stay with the same page and show the effect to the controlled system through the animated graphics parts of the page.

**IV.B. The Interactive Model Railroad**

We are using a model railroad layout in our laboratory as remote controlled system to demonstrate the capabilities of our system. The command evaluation system of our Web-server sends messages such as move train a to track 3. The realtime control system implemented on a PC will interpret the message, generate the commands to the switches, and start the appropriate model train through a digital command control system [Maerklin]. It will then poll sensors in the track to stop the train as soon as it arrives at the destination.

It should be noted that the feedback through the video transmission is not fast and accurate enough to fine-control the time critical process. It is however very suitable to diagnostic and planning tasks such as 'which train should go to which track'.

Encoding and decoding of GIF is fast enough to allow about 5 QCIF sized frames per second. We tested the performance in a local environment with a Sun workstation as server and Sun and Macintosh clients. Limiting factor in our demonstration scenario are the speed of the available frame grabber and the color conversions (dithering, colormap merging) for 8 bit pseudo color X-Window displays.

We do not exploit the frame differencing capabilities of GIF streams yet. We will add this feature to both the GIF encoder and WWW clients. We expect higher frame rates because encoder and decoder will have to process only the changed image parts. This will result in a speedup which is proportional to the relation between static and dynamic parts. The support of multiple clients makes the generation of the individual update information a complicated book-

**IV.C. A Commercial Application**

We are currently integrating the system with the World Wide Web front-end of a biochemical synthesis laboratory. Up to now clients of the laboratory have requested synthesis of oligonucleotides from remote via the WWW front-end. The synthesis has been done offline and the results with printed descriptions sent back via snail mail.

Soon clients will operate the robot from remote and watch the synthesis. The status will be displayed as a live video showing the equipment and as a graphics animation of the changing absorption spectrum. A client can modify synthesis parameters or even stop the process in case of problems. Of course the product has still to be sent with the postal service, but the turnaround times are decreased significantly.
V. Remarks and Conclusions

GIF uses LZW to compress bitmaps. UNISYS's LZW-copyright may lead to another common image format in the Web. We hope that a successor will support optimized encoding of image sequences as well. Candidates for a replacement of GIF are Planetary Data System (PDS) and Portable Network Graphics PNG [10], which is currently developed by Compuserve. Both specifications mention multi-image extensions.

We presented a scheme to include moving images into WWW pages without changing existing standards or protocols. The mechanism is based on the multi-image capabilities present in many image encoding standards and the HTTP protocol. In combination with the already standardized dialog elements of HTML, the system provides interactive video for the World Wide Web. Visual feedback to commands through animated graphics allows for a much smoother user interface than explicit confirmations. We validated the concepts with a sample implementation and the successful operation of a remote control scenario with inline video. The system is available and usable for every party (provider or consumer) on the internet.

VI. References


Figure 6: Remote WWW users can operate the model railroad and watch it in realtime.

The HTTP request shown in the picture is issued in order to get the contents of the inline image which is referenced by the base HTML document. The URI points to a stream synchronizing server extension. This server extension gets images from a live camera. The response to the HTTP request is an infinite image stream displayed at the client as video.

An HTML form is used to submit commands to the WWW server. The server forwards the commands to the model railroad controller via a serial interface. After hitting the 'Go!'-button the chosen train begins to move to the selected destination. An additional confirmation is not necessary.

http://rr-vs.informatik.uni-ulm.de/rr


Author Information

Konrad Froitzheim joined the Department of Distributed Systems at the University of Ulm, Germany in 1991. He spent 1990 at Farallon Computing Inc. in Emeryville/CA working on ISDN hardware and protocol software. Konrad Froitzheim received a doctorate from the University of Augsburg, Germany and a diploma in Mathematics from the Technical University in Munich, Germany. He has worked several years as software engineer developing business and office management applications.

Konrad Froitzheims research interests include distributed multimedia systems, computer and telecommunications, high performance networking, and mobile computing.

Heiner Wolf is a research assistant at the Department of Distributed Systems, University of Ulm, Germany. He received a diploma in physics from University of Freiburg, Germany and worked 1992 at CERN on parallel signal processing algorithms. Heiner Wolf joined the University of Ulm in 1993. His research interests include computer supported collaborative work (CSCW), realtime video compression, and communications.