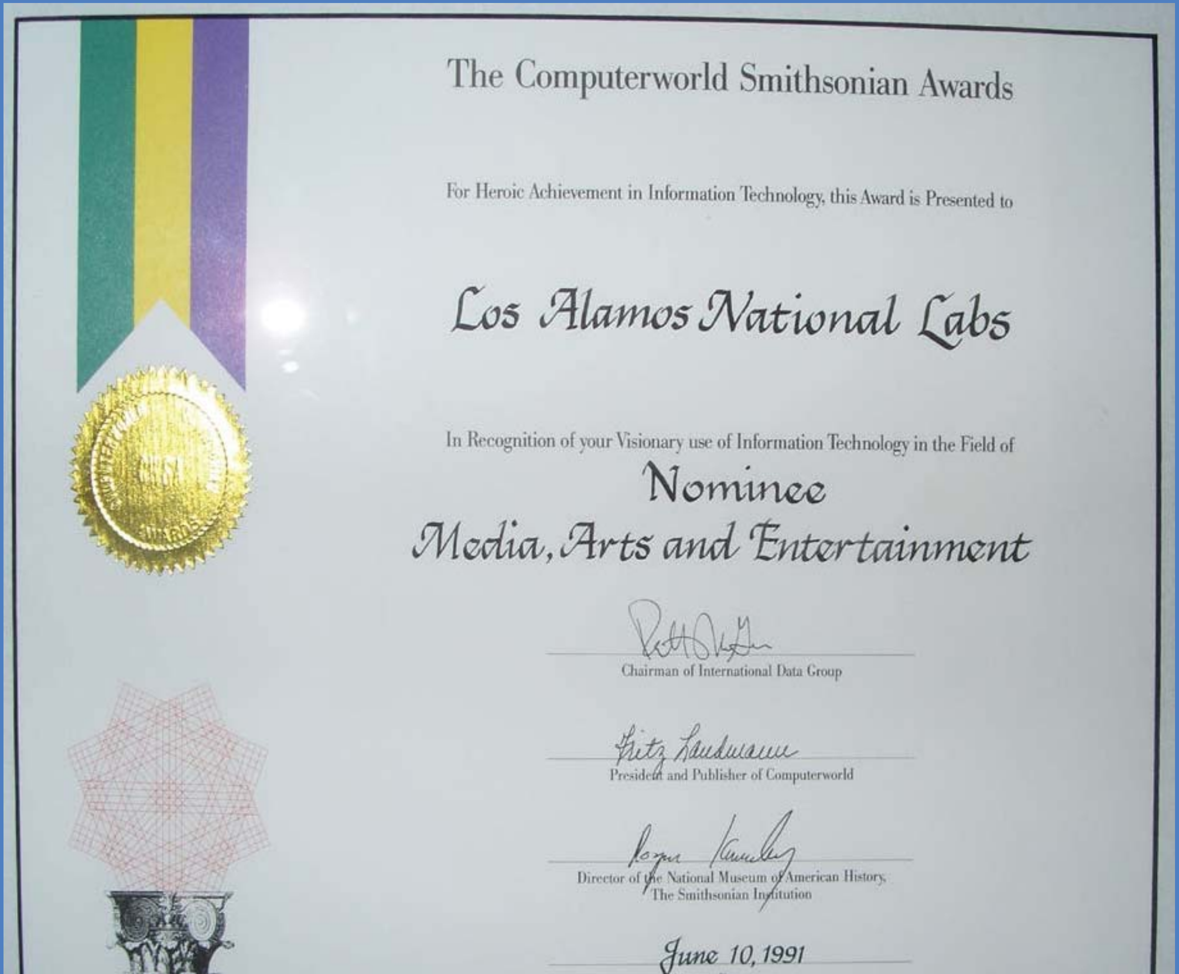
Exhibit 1

MediaView

I invented MediaView at Los Alamos in 1989

I was nominated for Smithsonian award for MediaVie



The Computerworld Smithsonian Awards

For Heroic Achievement in Information Technology, this Award is Presented to

Los Alamos National Labs

In Recognition of your Visionary use of Information Technology in the Field of

Nominee

Media, Arts and Entertainment

Chairman of International Data Group

Fritz Landmann
President and Publisher of Computerworld

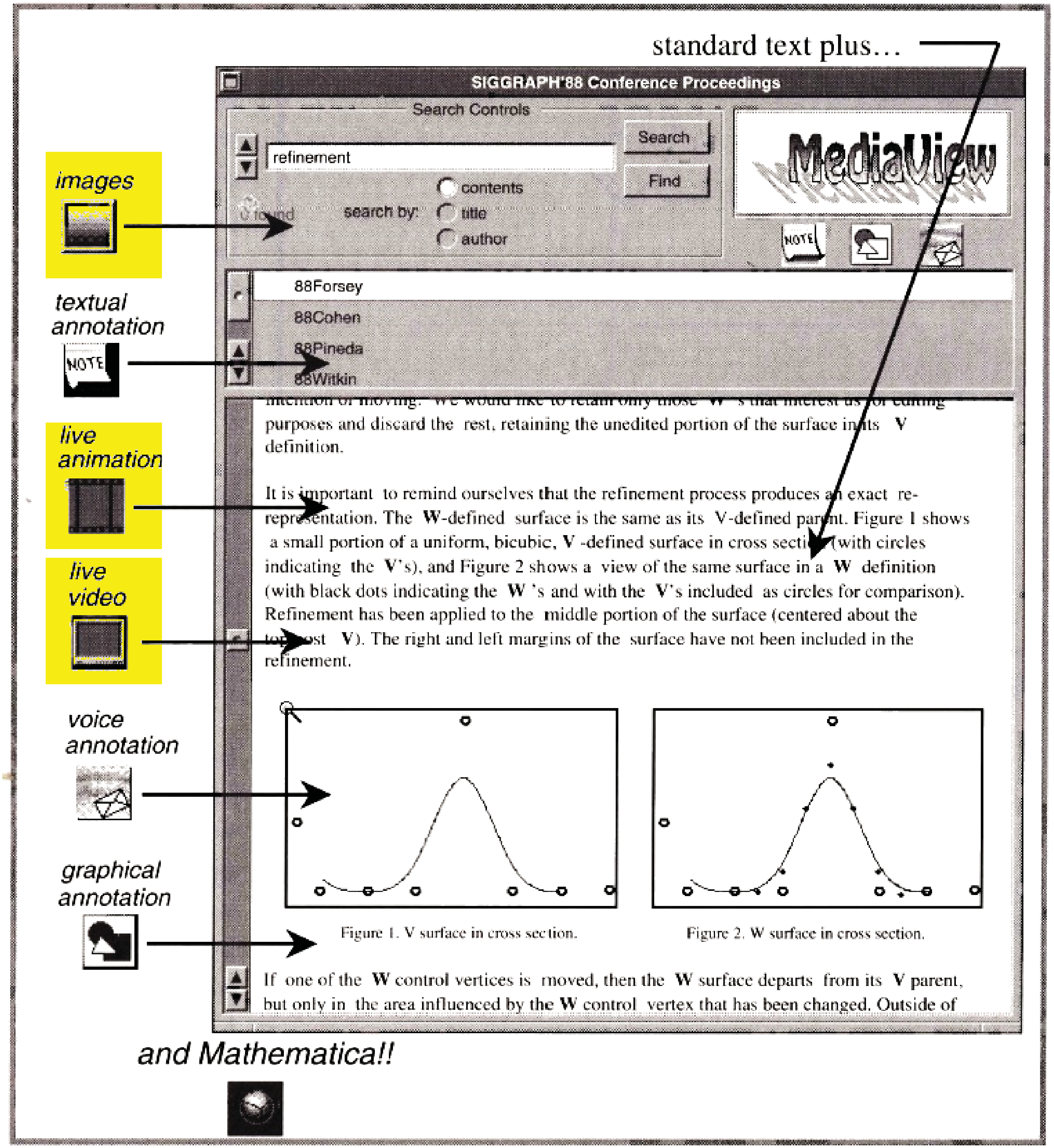
Director of the National Museum of American History,
The Smithsonian Institution

June 10, 1991

images

live animation

live video



intention of moving. We would like to retain only those W's that interest us for editing purposes and discard the rest, retaining the unedited portion of the surface in its V definition.

It is important to remind ourselves that the refinement process produces an exact re-representation. The W-defined surface is the same as its V-defined parent. Figure 1 shows a small portion of a uniform, bicubic, V -defined surface in cross section (with circles indicating the V's), and Figure 2 shows a view of the same surface in a W definition (with black dots indicating the W 's and with the V's included as circles for comparison). Refinement has been applied to the middle portion of the surface (centered about the topmost V). The right and left margins of the surface have not been included in the refinement.

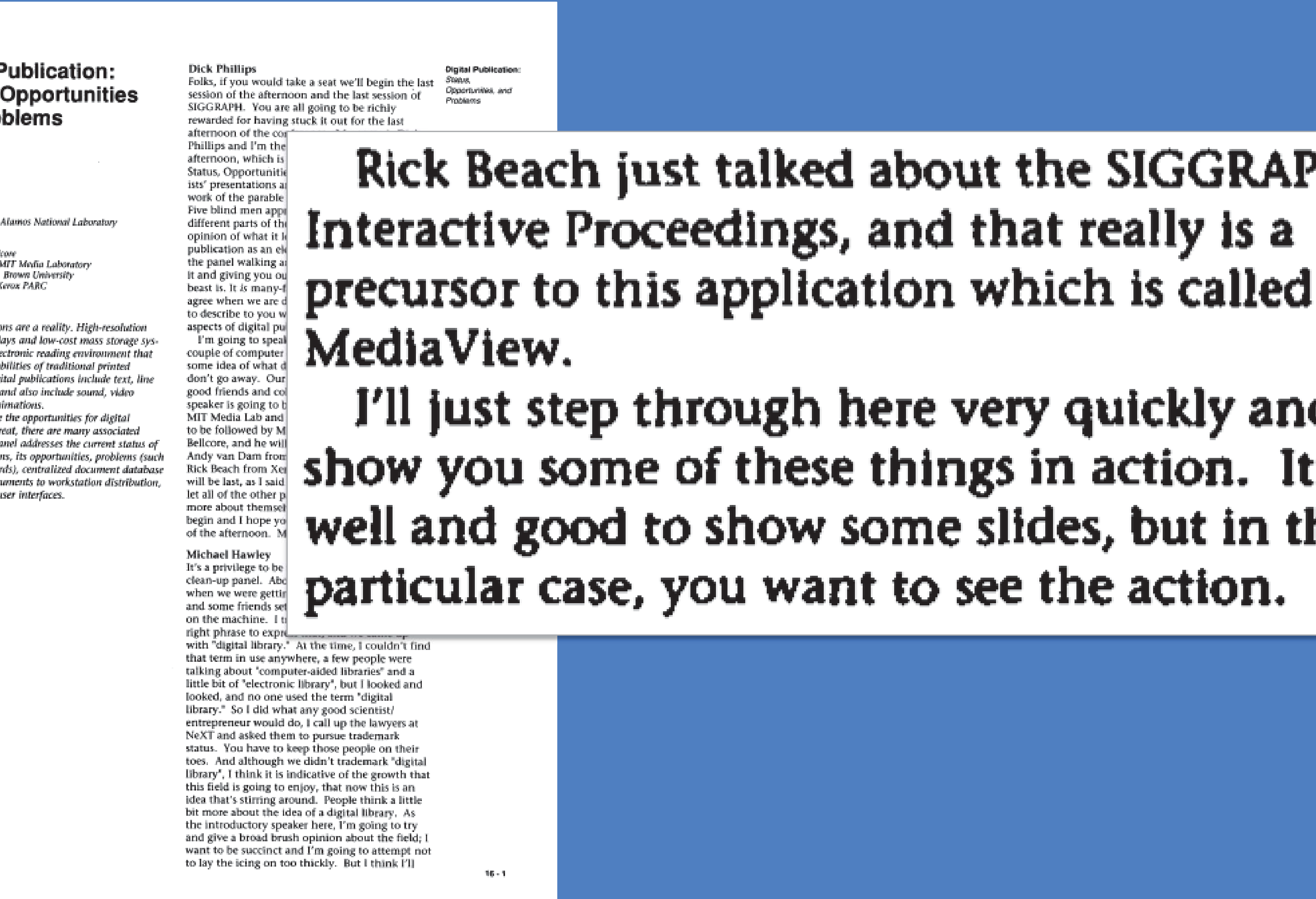
Figure 1. V surface in cross section.

Figure 2. W surface in cross section.

If one of the W control vertices is moved, then the W surface departs from its V parent, but only in the area influenced by the W control vertex that has been changed. Outside of

Figure 1. Components of a Media View document

Transcript of the 1990 SIGGRAPH conference in Dallas



Publication:
Opportunities
blems

Alamos National Laboratory

lcore
MIT Media Laboratory
Brown University
Xerox PARC

ons are a reality. High-resolution
lays and low-cost mass storage sys-
ectronic reading environment that
bilities of traditional printed
ital publications include text, line
and also include sound, video
imations.
e the opportunities for digital
reat, there are many associated
anel addresses the current status of
ns, its opportunities, problems (such
nds), centralized document database
uments to workstation distribution,
user interfaces.

Digital Publication:
Status,
Opportunities, and
Problems

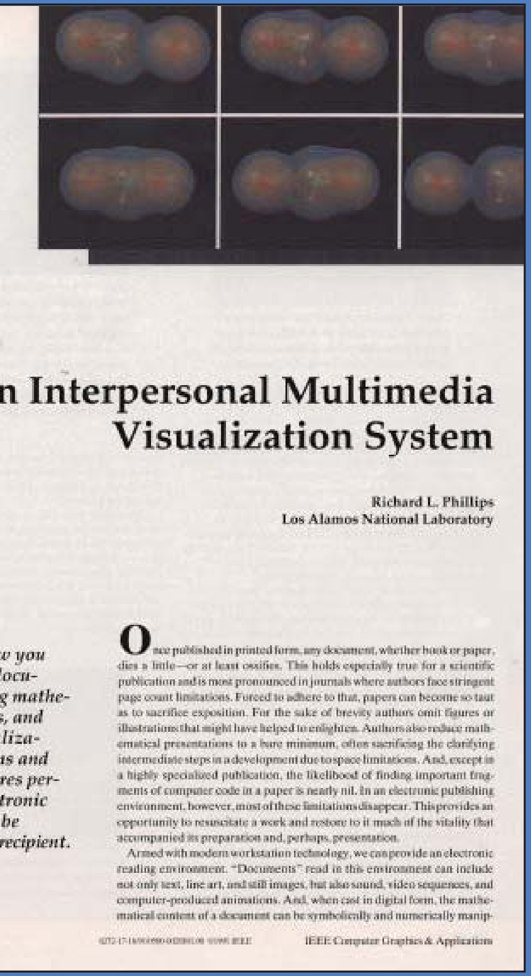
Dick Phillips
Folks, if you would take a seat we'll begin the last
session of the afternoon and the last session of
SIGGRAPH. You are all going to be richly
rewarded for having stuck it out for the last
afternoon of the co
Phillips and I'm the
afternoon, which is
Status, Opportuniti
ists' presentations a
work of the parable
Five blind men app
different parts of th
opinion of what it l
publication as an el
the panel walking a
it and giving you ou
beast is. It *is* many-f
agree when we are d
to describe to you w
aspects of digital pu
I'm going to speak
couple of computer
some idea of what d
don't go away. Our
good friends and co
speaker is going to b
MIT Media Lab and
to be followed by M
Bellcore, and he wil
Andy van Dam from
Rick Beach from Xe
will be last, as I said
let all of the other p
more about themsel
begin and I hope yo
of the afternoon. M

Michael Hawley
It's a privilege to be
clean-up panel. Abo
when we were getti
and some friends se
on the machine. I t
right phrase to expr
with "digital library." At the time, I couldn't find
that term in use anywhere, a few people were
talking about "computer-aided libraries" and a
little bit of "electronic library", but I looked and
looked, and no one used the term "digital
library." So I did what any good scientist/
entrepreneur would do, I call up the lawyers at
NeXT and asked them to pursue trademark
status. You have to keep those people on their
toes. And although we didn't trademark "digital
library", I think it is indicative of the growth that
this field is going to enjoy, that now this is an
idea that's stirring around. People think a little
bit more about the idea of a digital library. As
the introductory speaker here, I'm going to try
and give a broad brush opinion about the field; I
want to be succinct and I'm going to attempt not
to lay the icing on too thickly. But I think I'll

16 - 1

Rick Beach just talked about the SIGGRAP
Interactive Proceedings, and that really is a
precursor to this application which is called
MediaView.

I'll just step through here very quickly an
show you some of these things in action. It
well and good to show some slides, but in th
particular case, you want to see the action.



n Interpersonal Multimedia Visualization System

Richard L. Phillips
Los Alamos National Laboratory

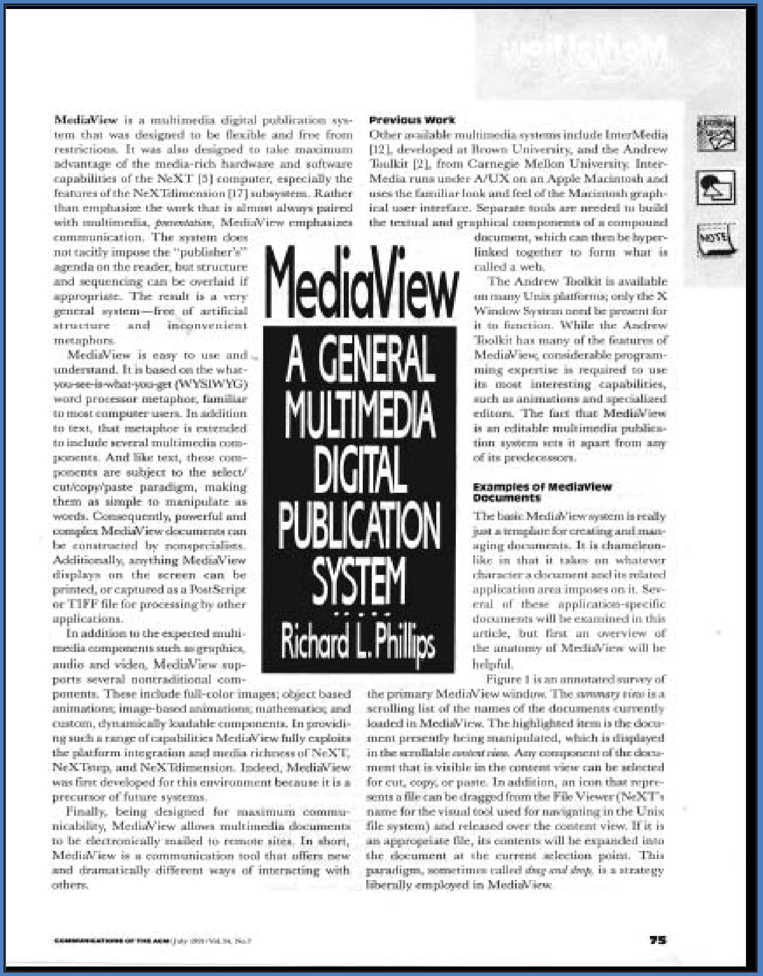
v you
locu-
g mathe-
s, and
liza-
ns and
res per-
tronic
be
recipient.

Once published in printed form, any document, whether book or paper, dies a little—or at least ossifies. This holds especially true for a scientific publication and is most pronounced in journals where authors face stringent page count limitations. Forced to adhere to that, papers can become so taut as to sacrifice exposition. For the sake of brevity authors omit figures or illustrations that might have helped to enlighten. Authors also reduce mathematical presentations to a bare minimum, often sacrificing the clarifying intermediate steps in a development due to space limitations. And, except in a highly specialized publication, the likelihood of finding important fragments of computer code in a paper is nearly nil. In an electronic publishing environment, however, most of these limitations disappear. This provides an opportunity to resuscitate a work and restore to it much of the vitality that accompanied its preparation and, perhaps, presentation.

Armed with modern workstation technology, we can provide an electronic reading environment. "Documents" read in this environment can include not only text, line art, and still images, but also sound, video sequences, and computer-produced animations. And, when cast in digital form, the mathematical content of a document can be symbolically and numerically manip-

0272-17-16/91/0900-0028$01.00 ©1991 IEEE IEEE Computer Graphics & Applications

1991 IEEE Article
JDX-005



MediaView A GENERAL MULTIMEDIA DIGITAL PUBLICATION SYSTEM

Richard L. Phillips

MediaView is a multimedia digital publication system that was designed to be flexible and free from restrictions. It was also designed to take maximum advantage of the media-rich hardware and software capabilities of the NeXT [5] computer, especially the features of the NeXTdimension [17] subsystem. Rather than emphasize the work that is almost always paired with multimedia, *presentation*, MediaView emphasizes communication. The system does not tacitly impose the "publisher's" agenda on the reader, but structure and sequencing can be overlaid if appropriate. The result is a very general system—free of artificial structure and inconvenient metaphors.

MediaView is easy to use and understand. It is based on the what-you-see-is-what-you-get (WYSIWYG) word processor metaphor, familiar to most computer users. In addition to text, that metaphor is extended to include several multimedia components. And like text, these components are subject to the select/cut/copy/paste paradigm, making them as simple to manipulate as words. Consequently, powerful and complex MediaView documents can be constructed by nonspecialists. Additionally, anything MediaView displays on the screen can be printed, or captured as a PostScript or TIFF file for processing by other applications.

In addition to the expected multimedia components such as graphics, audio and video, MediaView supports several nontraditional components. These include full-color images; object based animations; image-based animations; mathematics; and custom, dynamically loadable components. In providing such a range of capabilities MediaView fully exploits the platform integration and media richness of NeXT, NeXTstep, and NeXTdimension. Indeed, MediaView was first developed for this environment because it is a precursor of future systems.

Finally, being designed for maximum communicability, MediaView allows multimedia documents to be electronically mailed to remote sites. In short, MediaView is a communication tool that offers new and dramatically different ways of interacting with others.

Previous Work

Other available multimedia systems include InterMedia [12], developed at Brown University, and the Andrew Toolkit [2], from Carnegie Mellon University. InterMedia runs under A/UX on an Apple Macintosh and uses the familiar look and feel of the Macintosh graphical user interface. Separate tools are needed to build the textual and graphical components of a compound document, which can then be hyperlinked together to form what is called a web.

The Andrew Toolkit is available on many Unix platforms; only the X Window System need be present for it to function. While the Andrew Toolkit has many of the features of MediaView, considerable programming expertise is required to use its most interesting capabilities, such as animations and specialized editors. The fact that MediaView is an editable multimedia publication system sets it apart from any of its predecessors.

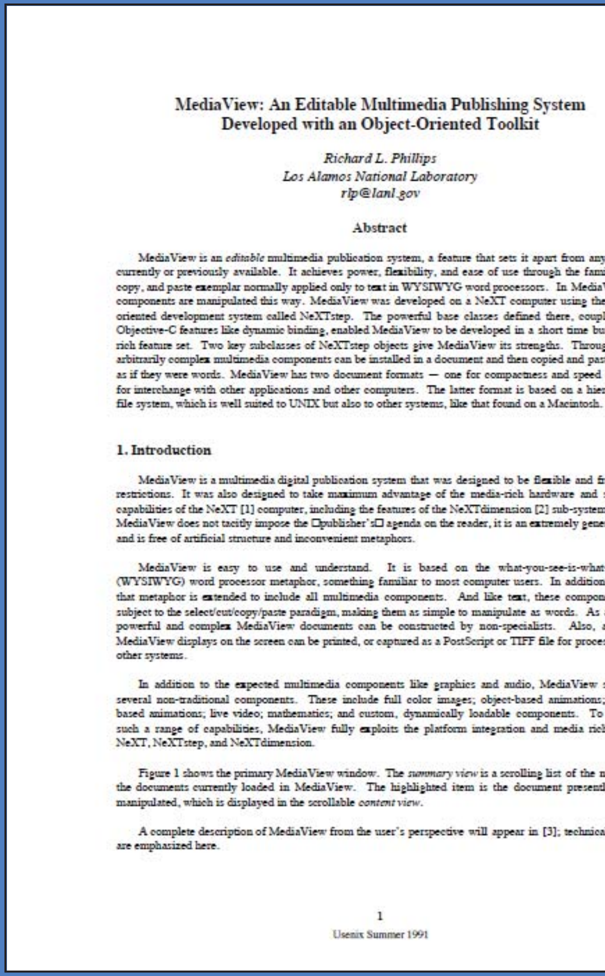
Examples of MediaView Documents

The basic MediaView system is really just a template for creating and managing documents. It is chameleon-like in that it takes on whatever character a document and its related application area imposes on it. Several of these application-specific documents will be examined in this article, but first an overview of the anatomy of MediaView will be helpful.

Figure 1 is an annotated survey of the primary MediaView window. The *summary view* is a scrolling list of the names of the documents currently loaded in MediaView. The highlighted item is the document presently being manipulated, which is displayed in the scrollable *content view*. Any component of the document that is visible in the content view can be selected for cut, copy, or paste. In addition, an icon that represents a file can be dragged from the File Viewer (NeXT's name for the visual tool used for navigating in the Unix file system) and released over the content view. If it is an appropriate file, its contents will be expanded into the document at the current selection point. This paradigm, sometimes called *drag and drop*, is a strategy liberally employed in MediaView.

COMMUNICATIONS OF THE ACM July 1991/Vol.34, No.7 75

1991 ACM Article
JDX-006



MediaView: An Editable Multimedia Publishing System Developed with an Object-Oriented Toolkit

Richard L. Phillips
Los Alamos National Laboratory
rlp@lanl.gov

Abstract

MediaView is an *editable* multimedia publication system, a feature that sets it apart from any
currently or previously available. It achieves power, flexibility, and ease of use through the fami
copy, and paste exemplar normally applied only to text in WYSIWYG word processors. In MediaV
components are manipulated this way. MediaView was developed on a NeXT computer using the
oriented development system called NeXTstep. The powerful base classes defined there, coupl
Objective-C features like dynamic binding, enabled MediaView to be developed in a short time bu
rich feature set. Two key subclasses of NeXTstep objects give MediaView its strengths. Throug
arbitrarily complex multimedia components can be installed in a document and then copied and pas
as if they were words. MediaView has two document formats — one for compactness and speed
for interchange with other applications and other computers. The latter format is based on a hier
file system, which is well suited to UNIX but also to other systems, like that found on a Macintosh.

1. Introduction

MediaView is a multimedia digital publication system that was designed to be flexible and fr
restrictions. It was also designed to take maximum advantage of the media-rich hardware and s
capabilities of the NeXT [1] computer, including the features of the NeXTdimension [2] sub-system
MediaView does not tacitly impose the □publisher's□ agenda on the reader, it is an extremely gene
and is free of artificial structure and inconvenient metaphors.

MediaView is easy to use and understand. It is based on the what-you-see-is-what
(WYSIWYG) word processor metaphor, something familiar to most computer users. In addition
that metaphor is extended to include all multimedia components. And like text, these compon
subject to the select/cut/copy/paste paradigm, making them as simple to manipulate as words. As
powerful and complex MediaView documents can be constructed by non-specialists. Also, a
MediaView displays on the screen can be printed, or captured as a PostScript or TIFF file for proces
other systems.

In addition to the expected multimedia components like graphics and audio, MediaView s
several non-traditional components. These include full color images; object-based animations;
based animations; live video; mathematics; and custom, dynamically loadable components. To
such a range of capabilities, MediaView fully exploits the platform integration and media rich
NeXT, NeXTstep, and NeXTdimension.

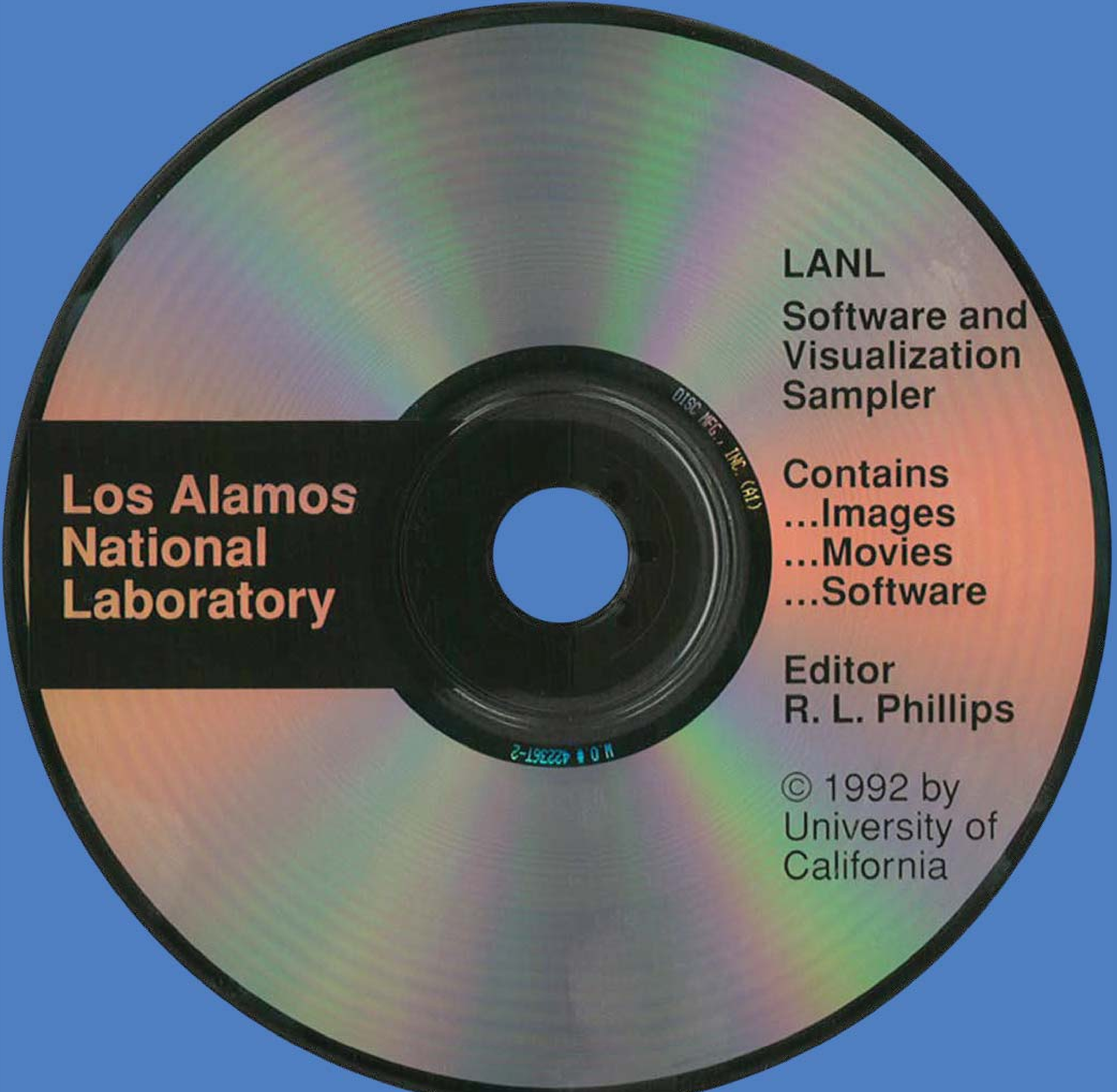
Figure 1 shows the primary MediaView window. The *summary view* is a scrolling list of the n
the documents currently loaded in MediaView. The highlighted item is the document presentl
manipulated, which is displayed in the scrollable *content view*.

A complete description of MediaView from the user's perspective will appear in [3]; technica
are emphasized here.

1

Usenix Summer 1991

1991 Usenix Article
JDX-007



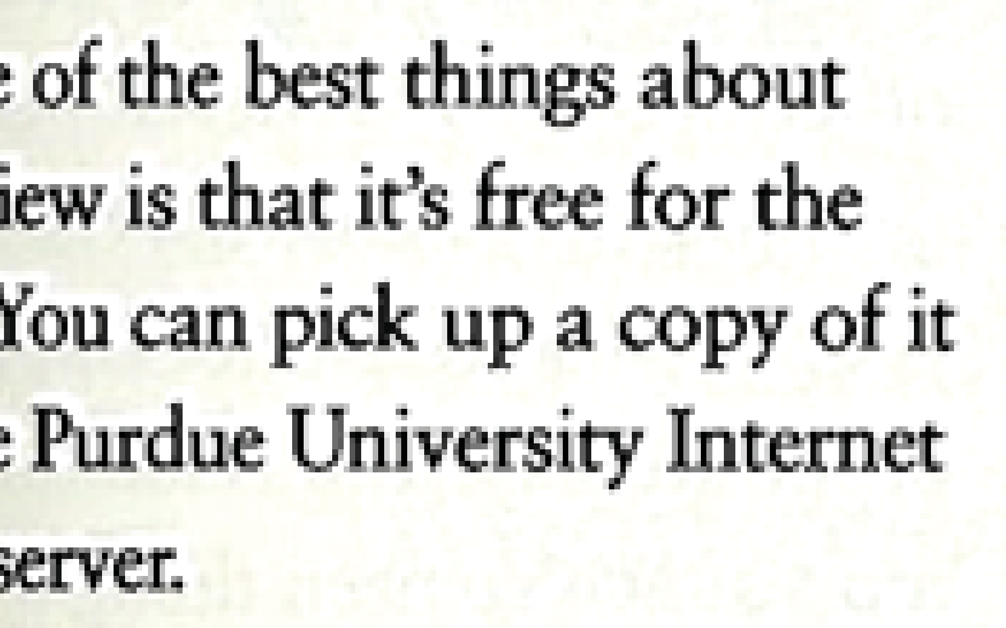
Los Alamos
National
Laboratory

LANL

Software and
Visualization
Sampler

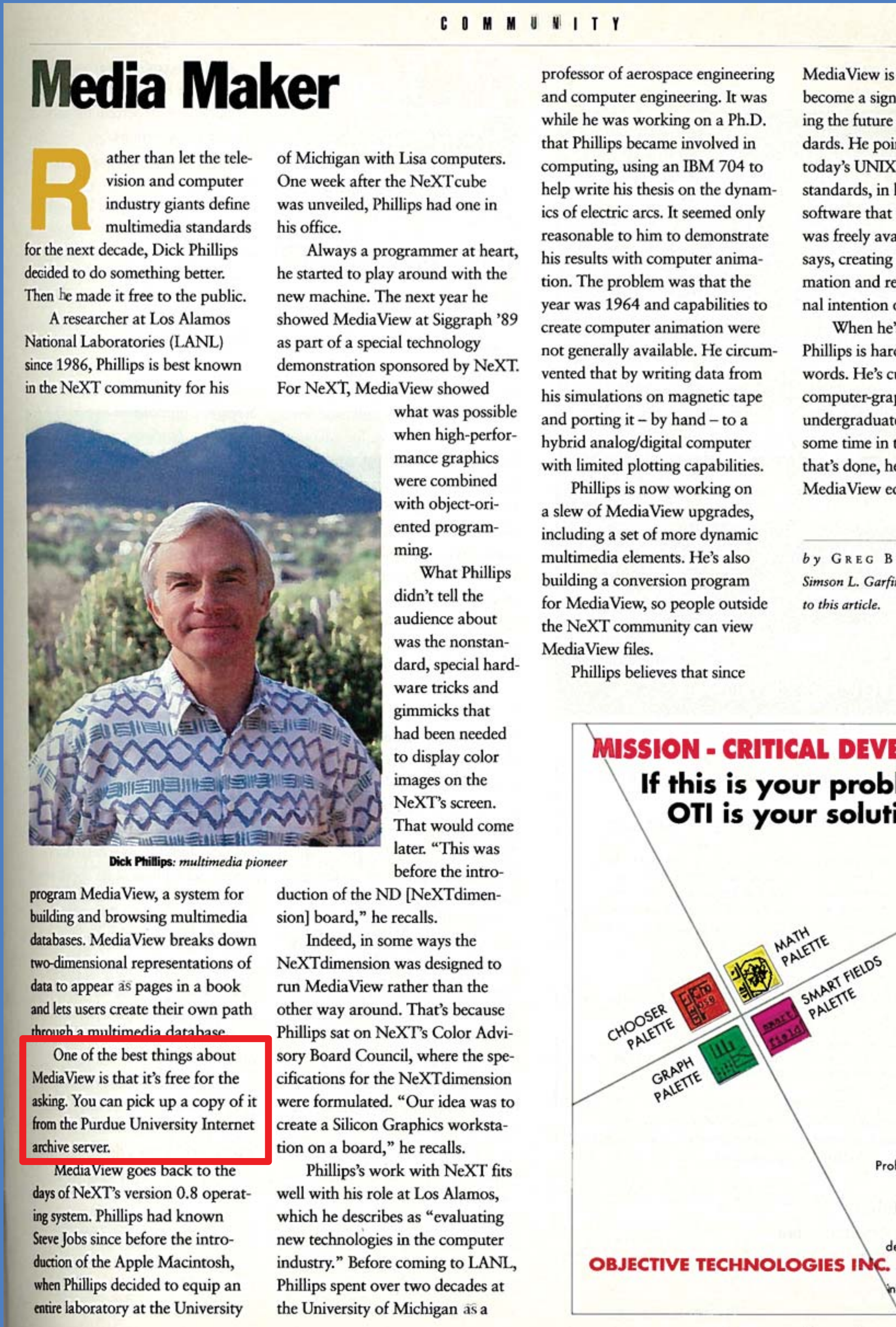
Contains
...Images
...Movies
...Software

Editor
R. L. Phillips



e of the best things about
iew is that it's free for the
You can pick up a copy of it
e Purdue University Internet
server.

Fall 1992 NEXTWORLD Article



Media Maker

Rather than let the television and computer industry giants define multimedia standards for the next decade, Dick Phillips decided to do something better. Then he made it free to the public.

A researcher at Los Alamos National Laboratories (LANL) since 1986, Phillips is best known in the NeXT community for his program MediaView, a system for building and browsing multimedia databases. MediaView breaks down two-dimensional representations of data to appear as pages in a book and lets users create their own path through a multimedia database.

Dick Phillips: *multimedia pioneer*

One of the best things about MediaView is that it's free for the asking. You can pick up a copy of it from the Purdue University Internet archive server.

MediaView goes back to the days of NeXT's version 0.8 operating system. Phillips had known Steve Jobs since before the introduction of the Apple Macintosh, when Phillips decided to equip an entire laboratory at the University of Michigan with Lisa computers. One week after the NeXTcube was unveiled, Phillips had one in his office.

Always a programmer at heart, he started to play around with the new machine. The next year he showed MediaView at Siggraph '89 as part of a special technology demonstration sponsored by NeXT. For NeXT, MediaView showed what was possible when high-performance graphics were combined with object-oriented programming.

What Phillips didn't tell the audience about was the nonstandard, special hardware tricks and gimmicks that had been needed to display color images on the NeXT's screen. That would come later. "This was before the introduction of the ND [NeXTdimension] board," he recalls.

Indeed, in some ways the NeXTdimension was designed to run MediaView rather than the other way around. That's because Phillips sat on NeXT's Color Advisory Board Council, where the specifications for the NeXTdimension were formulated. "Our idea was to create a Silicon Graphics workstation on a board," he recalls.

Phillips's work with NeXT fits well with his role at Los Alamos, which he describes as "evaluating new technologies in the computer industry." Before coming to LANL, Phillips spent over two decades at the University of Michigan as a professor of aerospace engineering and computer engineering. It was while he was working on a Ph.D. that Phillips became involved in computing, using an IBM 704 to help write his thesis on the dynamics of electric arcs. It seemed only reasonable to him to demonstrate his results with computer animation. The problem was that the year was 1964 and capabilities to create computer animation were not generally available. He circumvented that by writing data from his simulations on magnetic tape and porting it – by hand – to a hybrid analog/digital computer with limited plotting capabilities.

Phillips is now working on a slew of MediaView upgrades, including a set of more dynamic multimedia elements. He's also building a conversion program for MediaView, so people outside the NeXT community can view MediaView files.

Phillips believes that since

MediaView is
become a sign
ing the future
dards. He poi
today's UNIX
standards, in
software that
was freely ava
says, creating
mation and re
nal intention

When he'
Phillips is har
words. He's c
computer-gra
undergraduat
some time in
that's done, h
MediaView e

by Greg B
Simson L. Garfi
to this article.

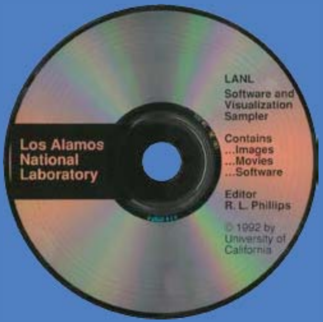
n

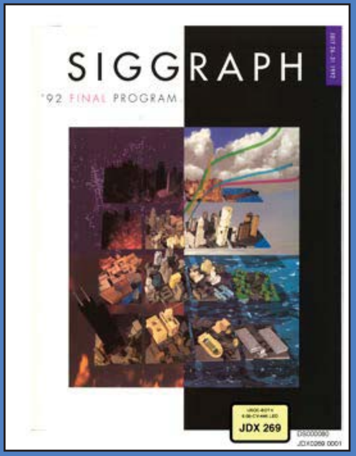
1991
three articles
published
about MediaView

MediaView: An Editable Multimedia Publishing System
Developed with an Object-Oriented Toolkit
MediaView
A GENERAL
MULTIMEDIA
DIGITAL
PUBLICATION
SYSTEM
Richard L. Phillips

1992
MediaView CD publicly
available

MediaView featured in
magazine as available for
free download

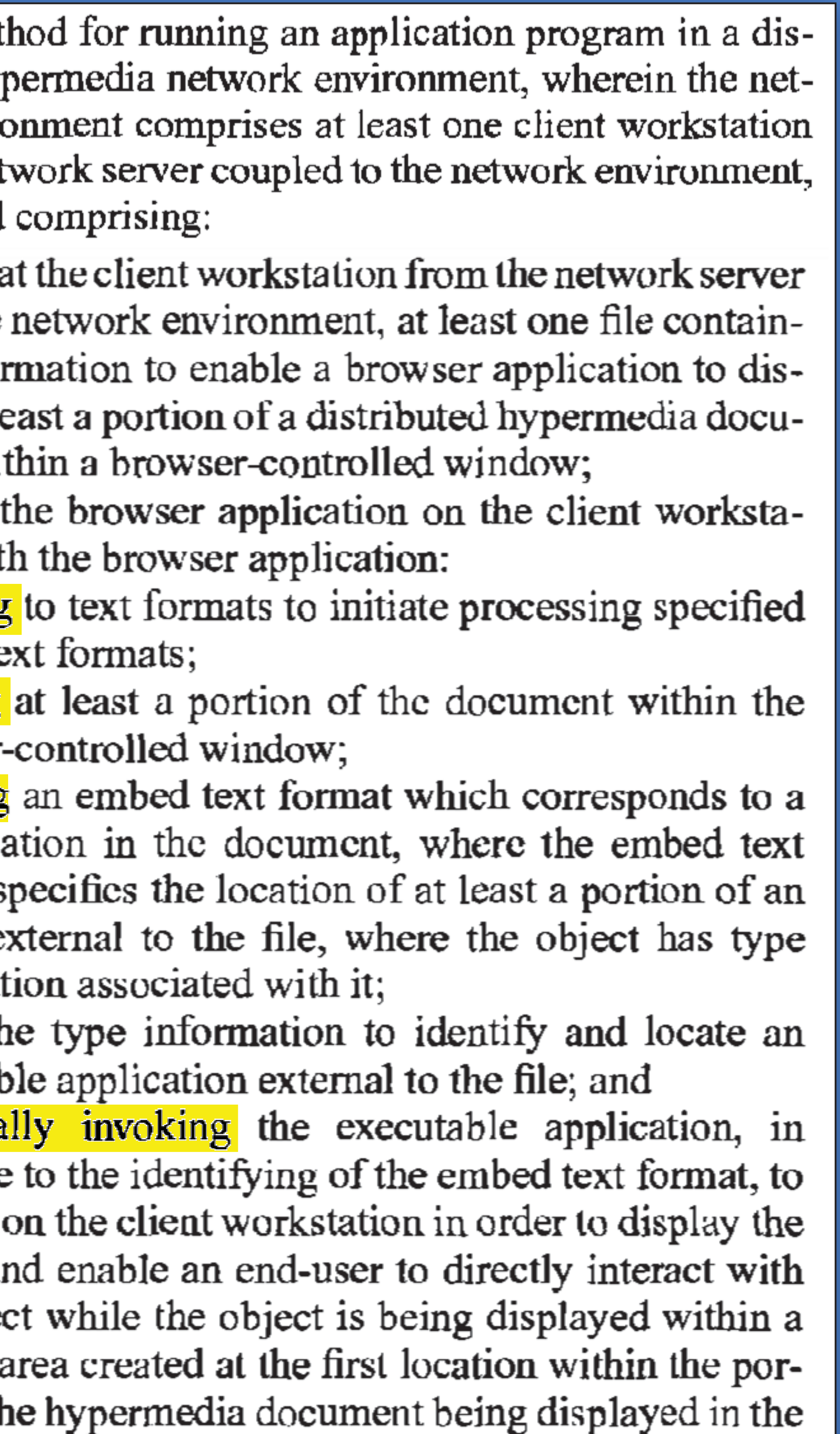




O
’9

Demonstrates MediaView 2, which pre-dates Eolas Patents

MediaView was NOT considered by the Patent Office



thod for running an application program in a dis-
permedia network environment, wherein the net-
onment comprises at least one client workstation
twork server coupled to the network environment,
l comprising:

at the client workstation from the network server
e network environment, at least one file contain-
rmation to enable a browser application to dis-
east a portion of a distributed hypermedia docu-
thin a browser-controlled window;

the browser application on the client worksta-
th the browser application:

g to text formats to initiate processing specified
ext formats;

at least a portion of the document within the
-controlled window;

g an embed text format which corresponds to a
ation in the document, where the embed text
specifics the location of at least a portion of an
xternal to the file, where the object has type
tion associated with it;

he type information to identify and locate an
ble application external to the file; and

ally invoking the executable application, in
e to the identifying of the embed text format, to
on the client workstation in order to display the
nd enable an end-user to directly interact with
ct while the object is being displayed within a
area created at the first location within the por-
he hypermedia document being displayed in the

1 receiving …

2 executing …

3 responding …

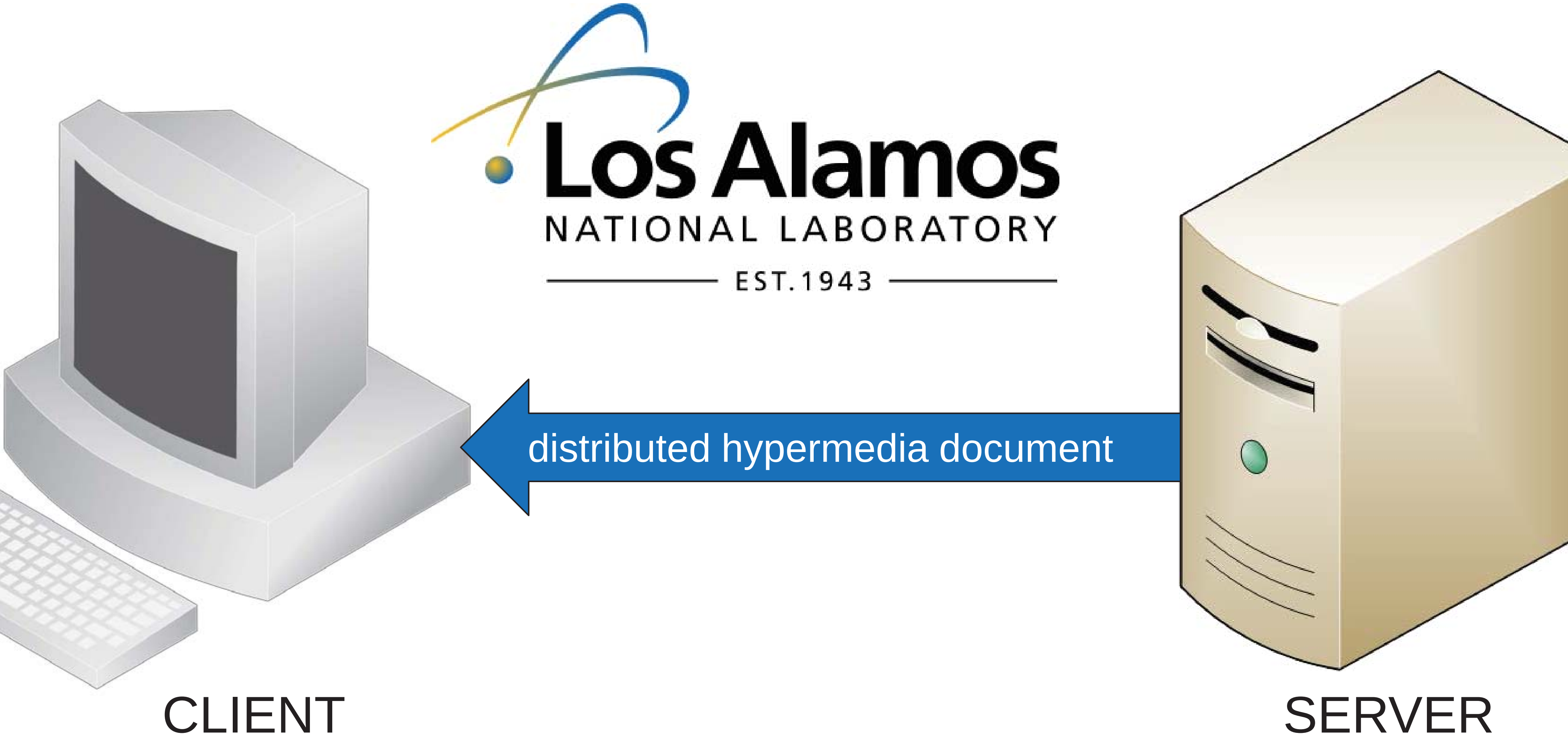
4 displaying …

5 identifying …

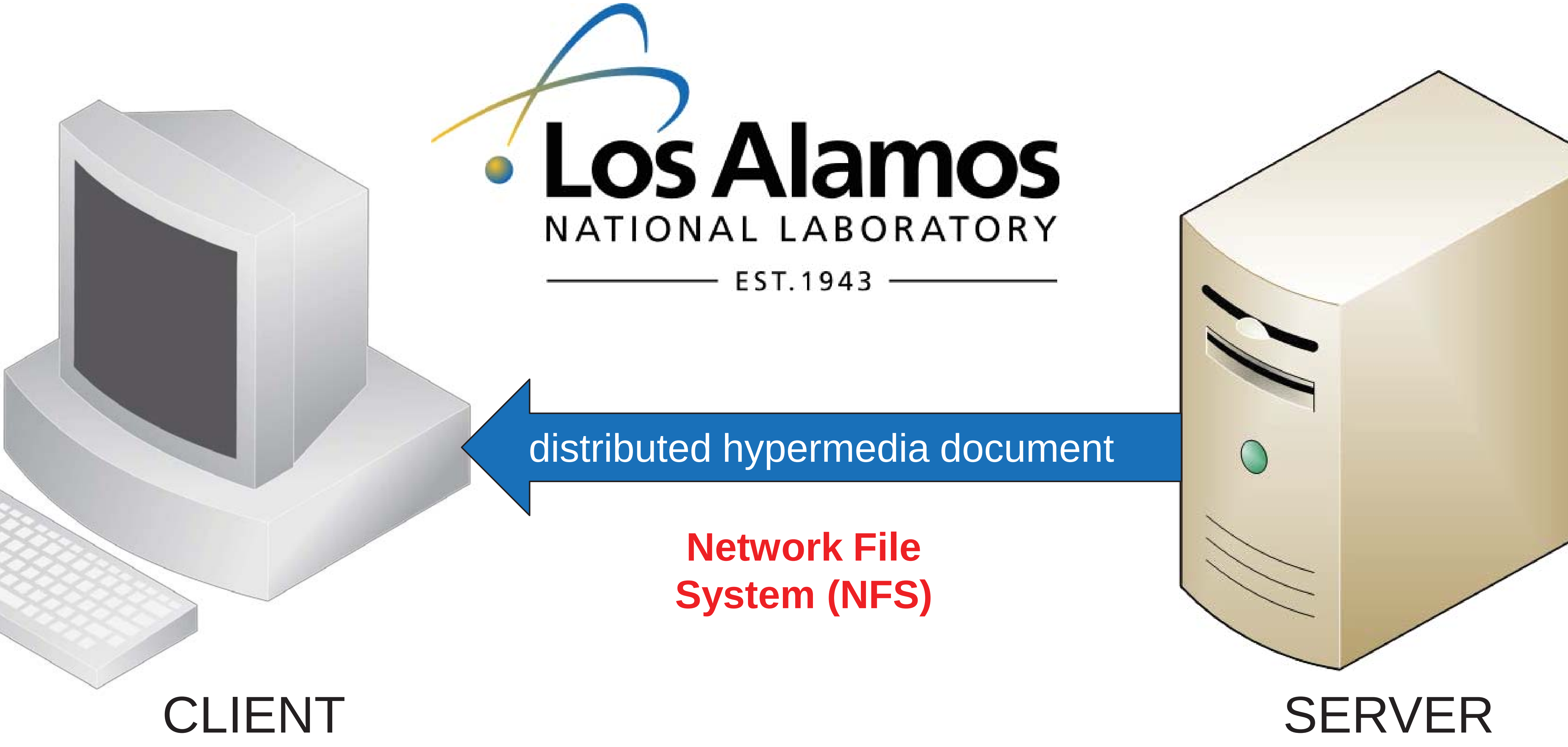
6 utilizing …

7 automatically invoki

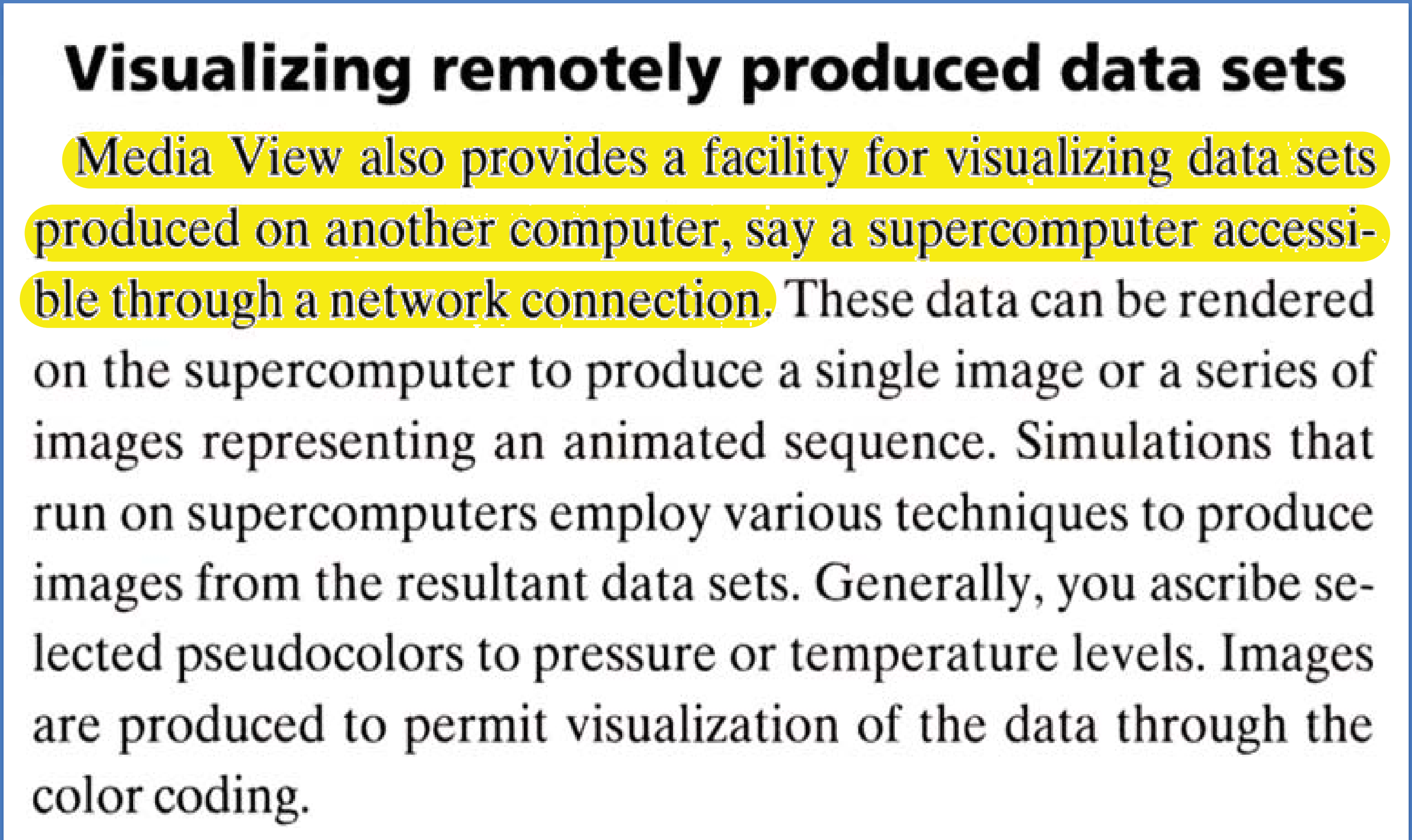
ethod for running an application program in a distributed **hypermedi** **work** environment, wherein the network environment comprises at le client workstation and one **network server** coupled to the network ironment, the method comprising:


Los Alamos NATIONAL LABORATORY EST.1943
distributed hypermedia document
CLIENT
SERVER

eiving, at the client workstation **from the network server over the work environment**, at least one file containing information to enable a wser application to display at least a portion of a distributed hypermed ument within a browser-controlled window;



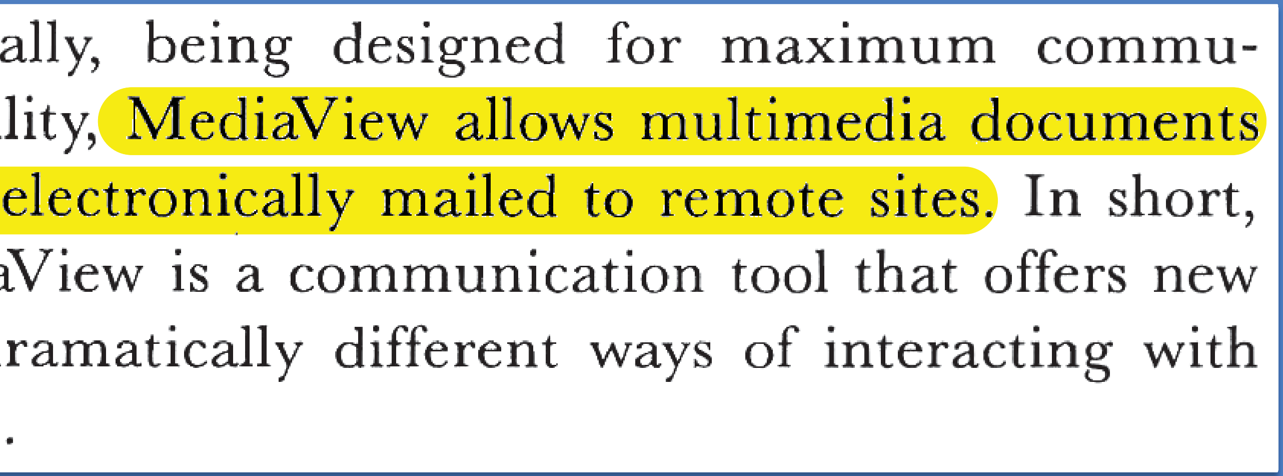
eiving, at the client workstation **from the network server over the work environment**, at least one file containing information to enable a wser application to display at least a portion of a distributed hypermed ument within a browser-controlled window;



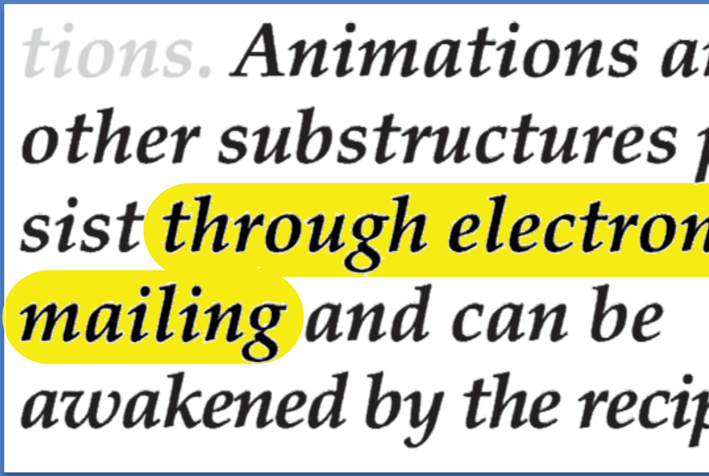
Visualizing remotely produced data sets

Media View also provides a facility for visualizing data sets produced on another computer, say a supercomputer accessible through a network connection. These data can be rendered on the supercomputer to produce a single image or a series of images representing an animated sequence. Simulations that run on supercomputers employ various techniques to produce images from the resultant data sets. Generally, you ascribe selected pseudocolors to pressure or temperature levels. Images are produced to permit visualization of the data through the color coding.

eiving, at the client workstation **from the network server over the work environment**, at least one file containing information to enable a wser application to display at least a portion of a distributed hypermed ument within a browser-controlled window;

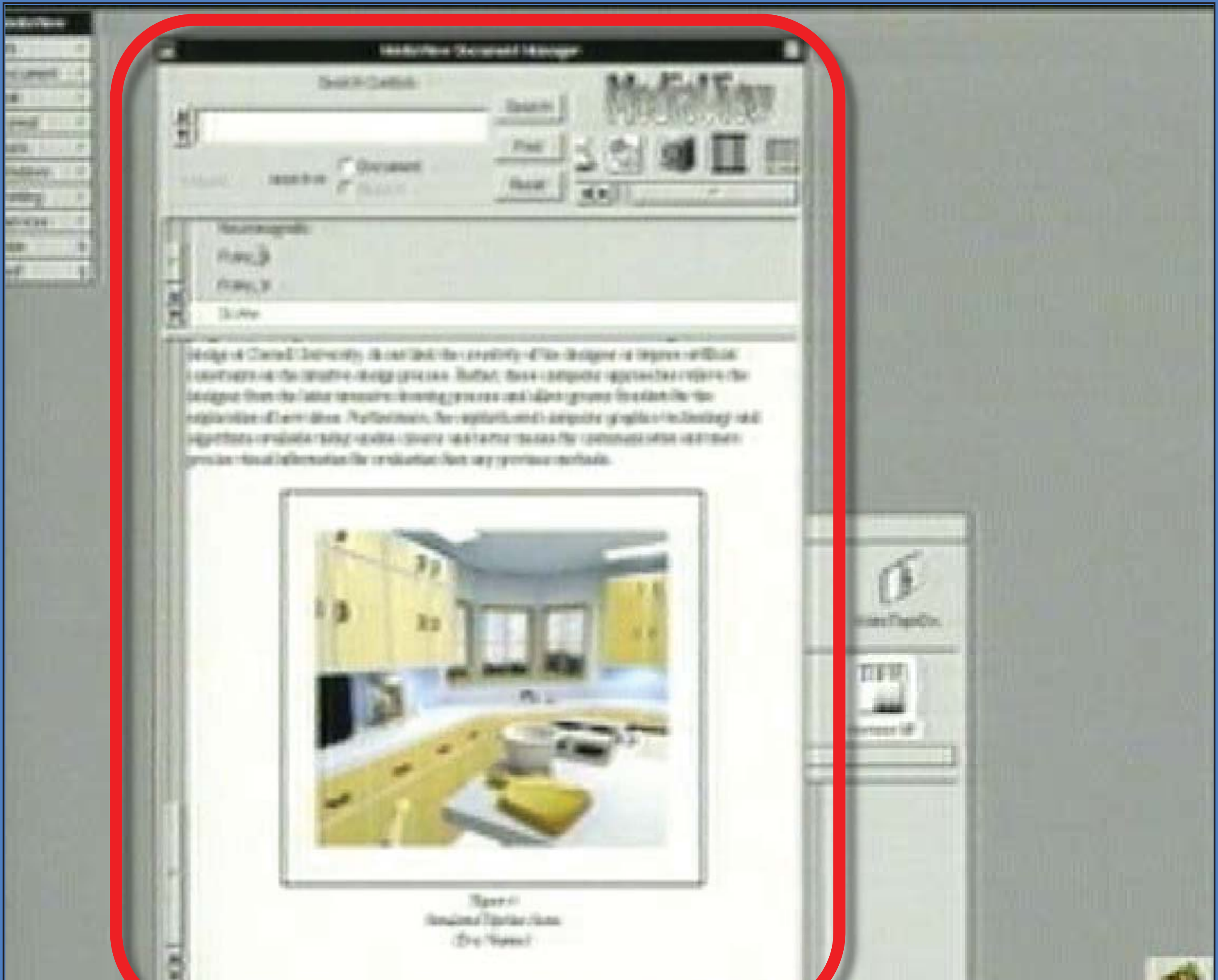

ally, being designed for maximum commu- lity, MediaView allows multimedia documents electronically mailed to remote sites. In short, aView is a communication tool that offers new ramatically different ways of interacting with .

1991 ACM Article: JDX 006

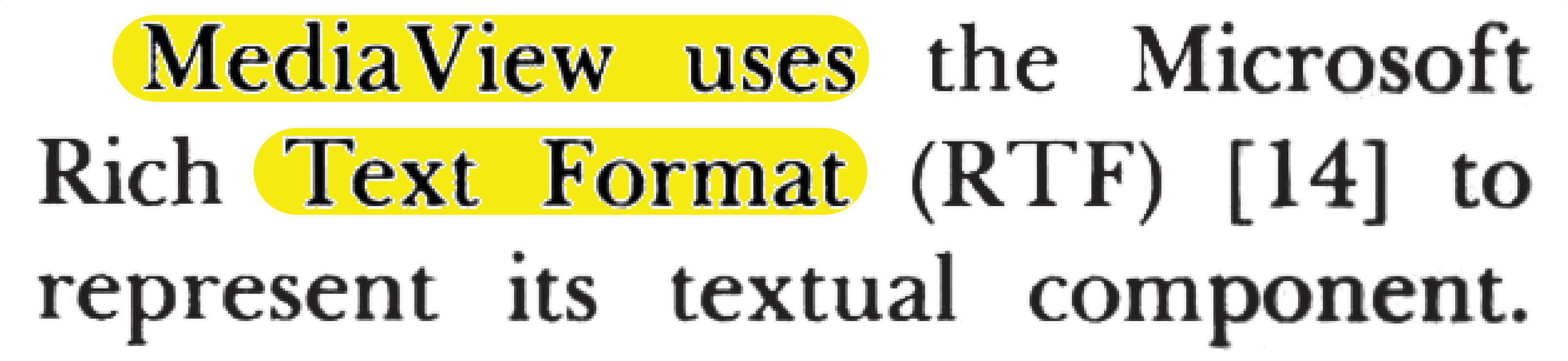

tions. Animations a other substructures p sist through electron mailing and can be awakened by the recip

1991 IEEE Article: J

cuting **the browser application on the client workstation**, with the
wser application:



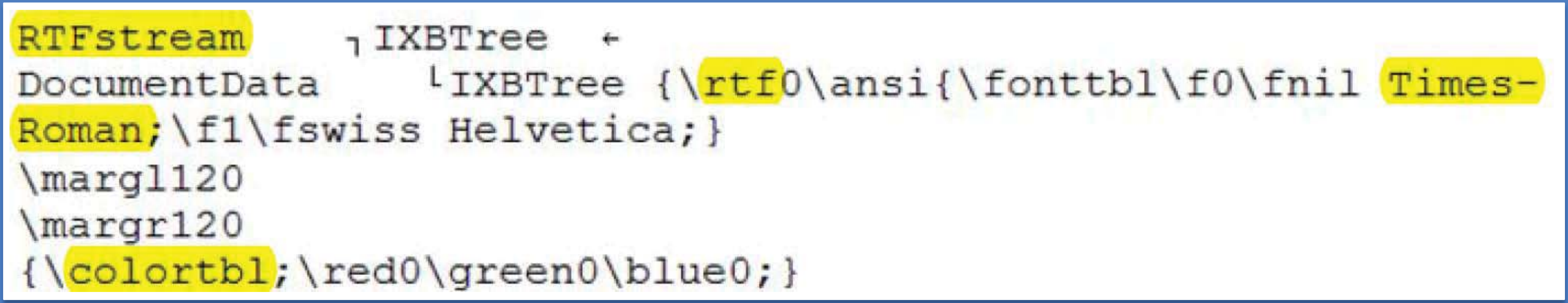
onding **to text formats** to initiate processing specified by the text fo



MediaView uses the Microsoft Rich Text Format (RTF) [14] to represent its textual component.

1991 ACM Article: JDX 006

ediaView Hypermedia File Excerpt:

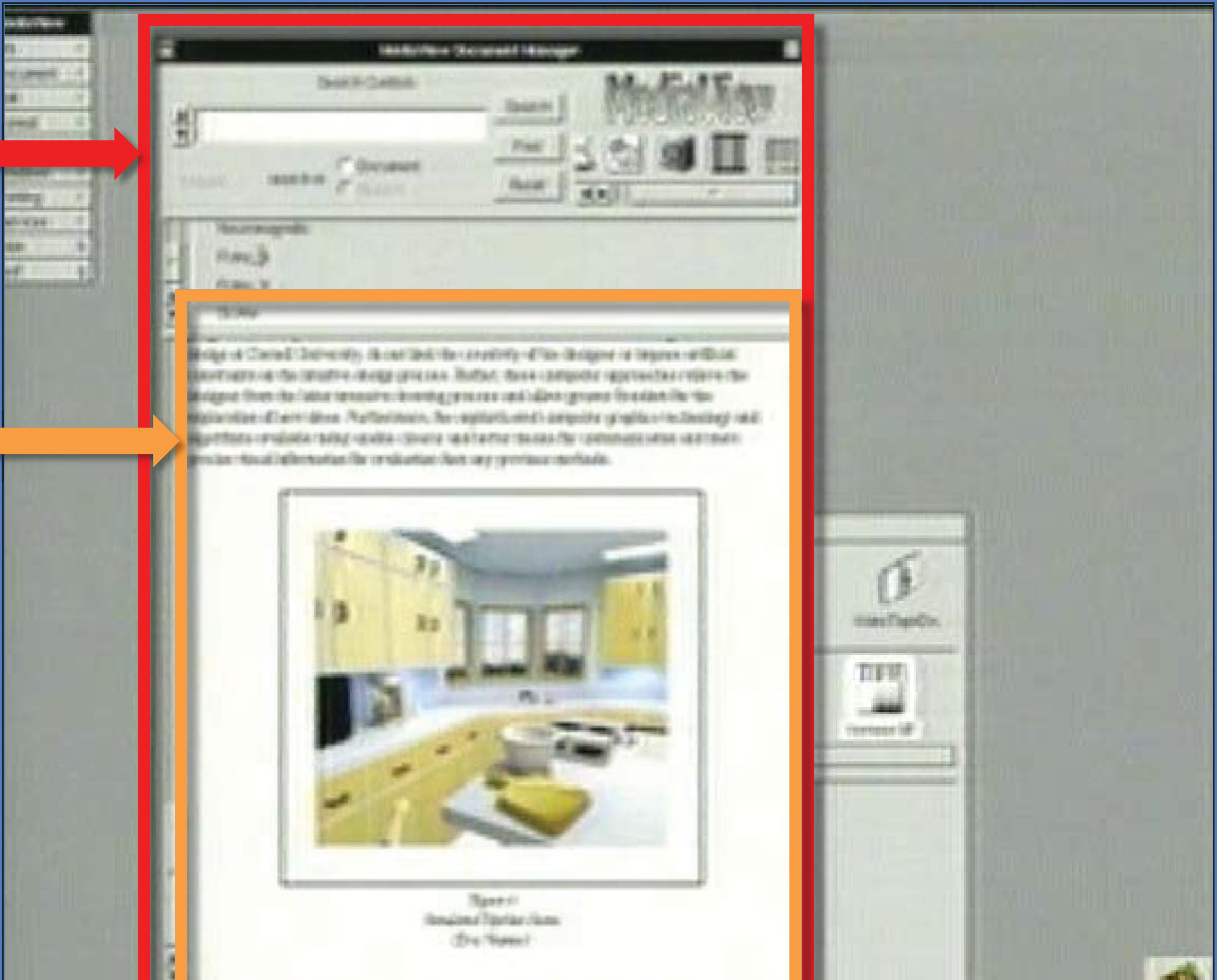


```
RTFstream      ┐IXBTree  ←
DocumentData     └IXBTree {\rtf0\ansi{\fonttbl\f0\fnil Times-
Roman;\f1\fswiss Helvetica;}
\margl120
\margr120
{\colortbl;\red0\green0\blue0;}
```

playing at least a portion of the document **within the browser-contro**
dow;

controlled window

n of the document

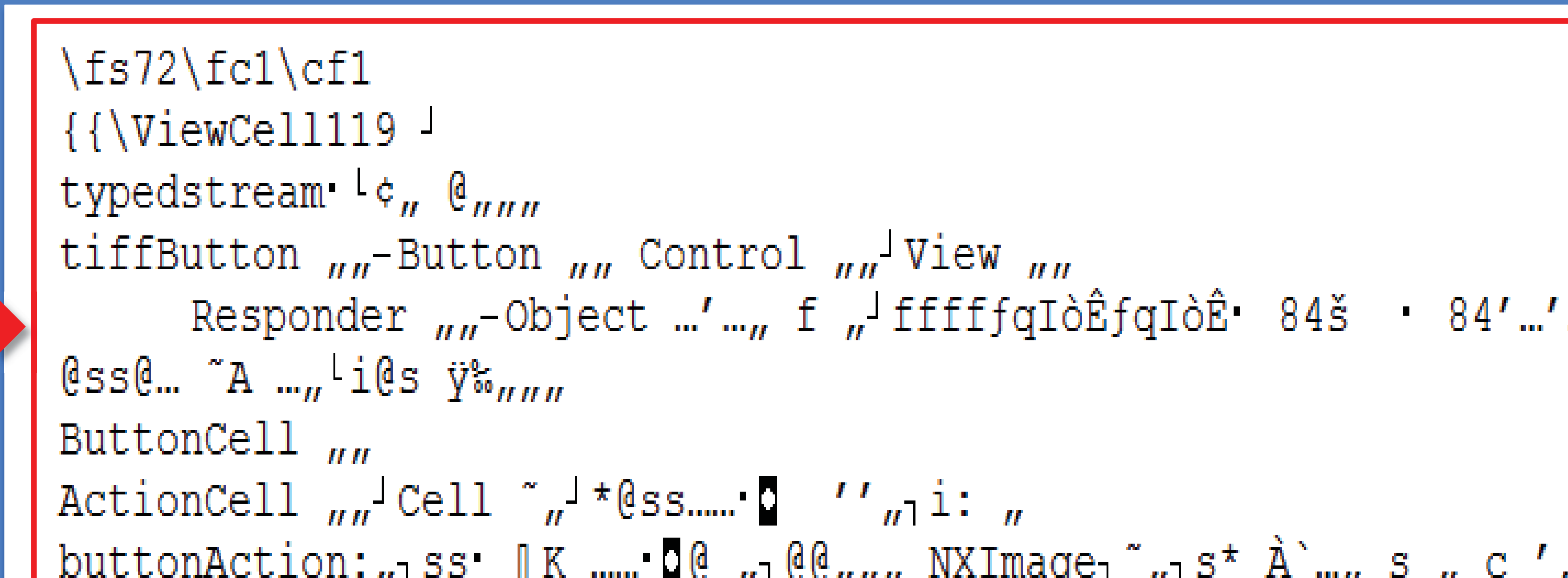


ntifying an **embed text format** which corresponds to a first location i
ument, where the embed text format specifies the location of at least
ion of an object external to the file, where the object has type informa
ociated with it;

rt on "embed text format": coded information that specifies to a browser
ication that an object is to be embedded in a displayed hypermedia docume

liaView Hypermedia File Excerpt:

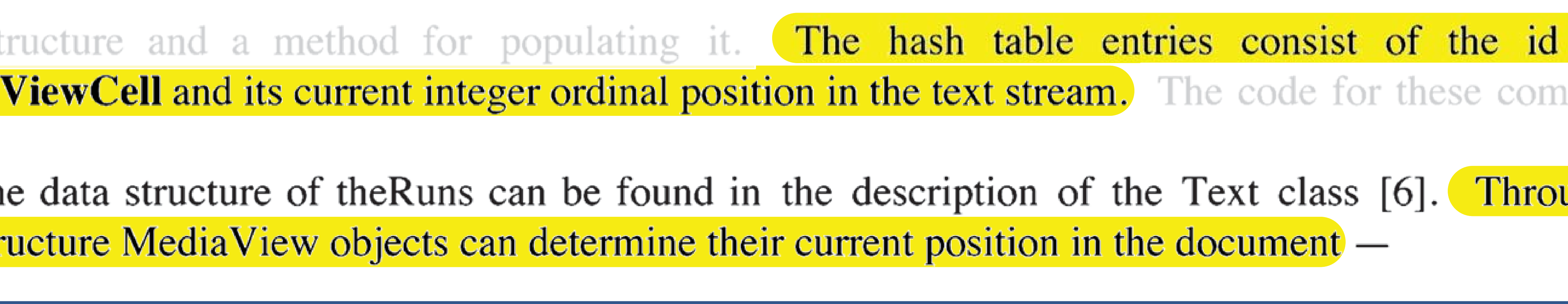
text format



```
\fs72\fc1\cf1
{{\ViewCell119 ┘
typedstream·└¢„ @„„„
tiffButton „„-Button „„ Control „„┘View „„
      Responder „„-Object …'…„ f „┘ffffqIòÊfqIòÊ· 84š  · 84'…'
@ss@… ˜A …„└i@s ÿ‰„„„
ButtonCell „„
ActionCell „„┘Cell ˜„┘*@ss……·■  ''„┐i: „
buttonAction:„┐ss· ▯K ……·■@ „┐@@„„„ NXImage┐˜„┐s* À`…„ s „ c '
```

ntifying an embed text format **which corresponds to a first locatior**
ument, where the embed text format specifies the location of at least
ion of an object external to the file, where the object has type informa
ociated with it;

cation of MediaView “object” “corresponds to” location of
mbedded text format”:

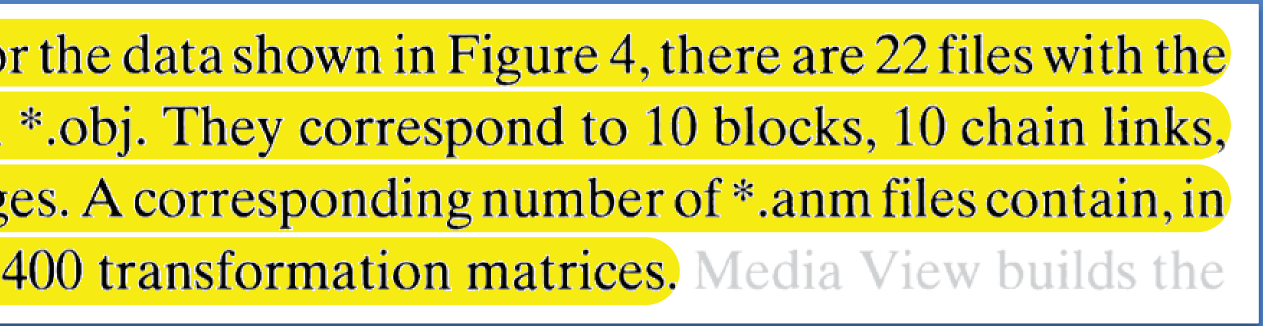


tructure and a method for populating it. The hash table entries consist of the id
ViewCell and its current integer ordinal position in the text stream. The code for these com

e data structure of theRuns can be found in the description of the Text class [6]. Throu
ucture MediaView objects can determine their current position in the document —

1991 Usenix Article

ntifying an embed text format which corresponds to a first location in ument, **where the embed text format specifies the location of at le tion of an object external to the file**, where the object has type infor ociated with it;



or the data shown in Figure 4, there are 22 files with the *.obj. They correspond to 10 blocks, 10 chain links, ges. A corresponding number of *.anm files contain, in 400 transformation matrices. Media View builds the

1991 IEEE Article: JDX 005

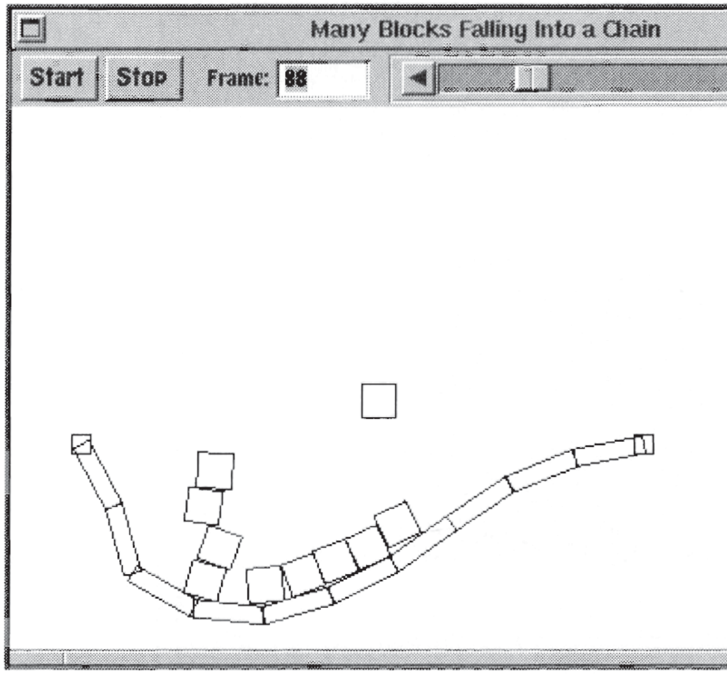


Figure 4. The Media View data set animation facil

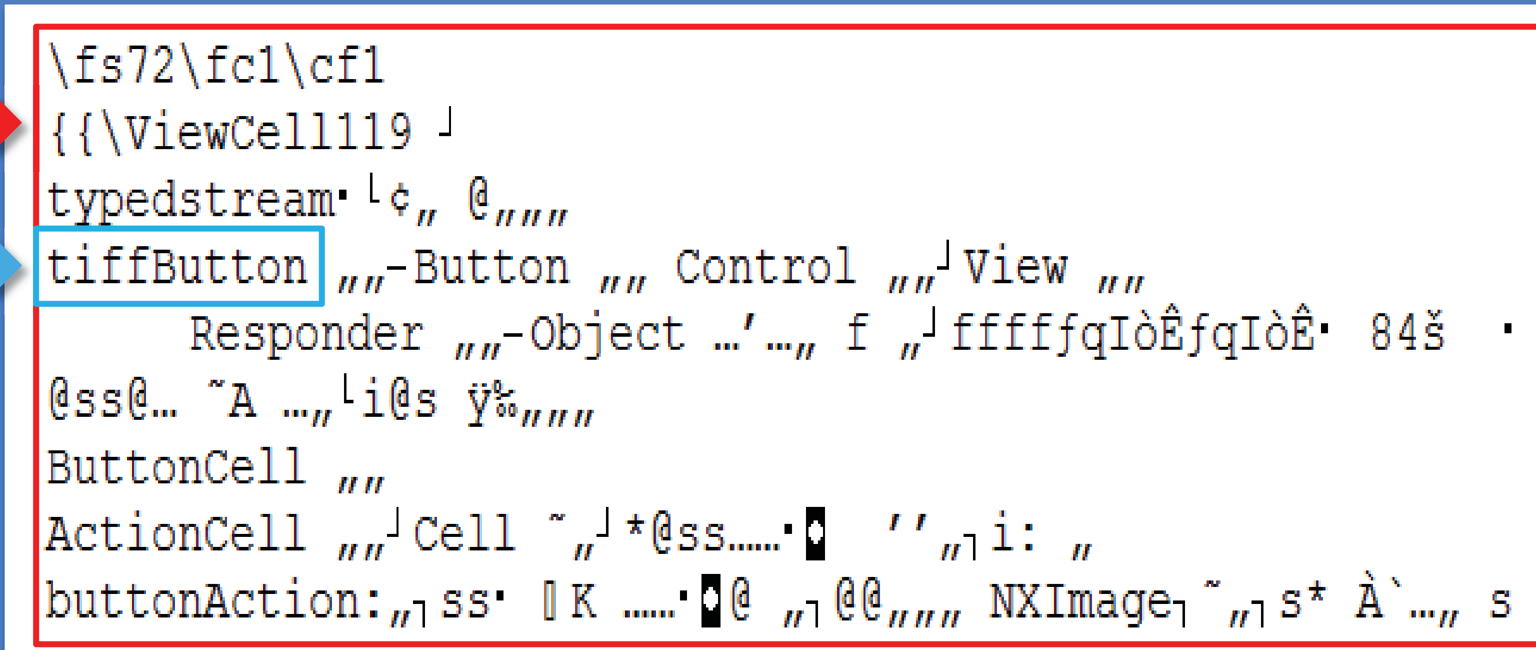
ntifying an embed text format which corresponds to a first location in
ument, where the embed text format specifies the location of at least
ion of an object external to the file, **where the object has type inforr**
ociated with it;

<u>rt on “object”</u>: Text, images, sound files, video data, documents, and/or othe
s of information that is presentable to a user of a computer system

liaView Hypermedia File Excerpt:

mbed text format

ation associated
thematica object



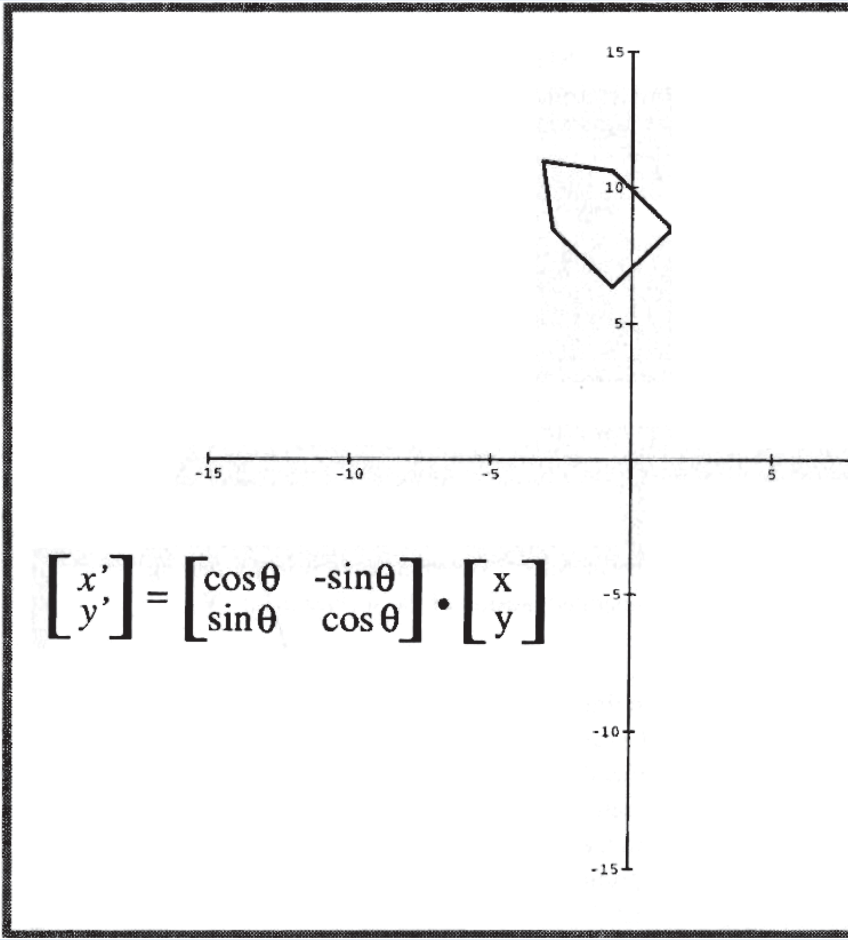
```
\fs72\fc1\cf1
{{\ViewCell119 ┘
typedstream·└¢„ @„„„
tiffButton „„–Button „„ Control „„┘View „„
      Responder „„–Object …′…„ f „┘ffffqIòÊfqIòÊ· 84š  ·
@ss@… ˜A …„└i@s ÿ‰„„„
ButtonCell „„
ActionCell „„┘Cell ˜„┘*@ss……·◘  ′′„┐i: „
buttonAction:„┐ss· ▯K ……·◘@ „┐@@„„„ NXImage┐˜„┐s* À`…„ s
```

zing the type information to identify and locate an **executable applic**
ernal to the file; and

Linking to Mathematica

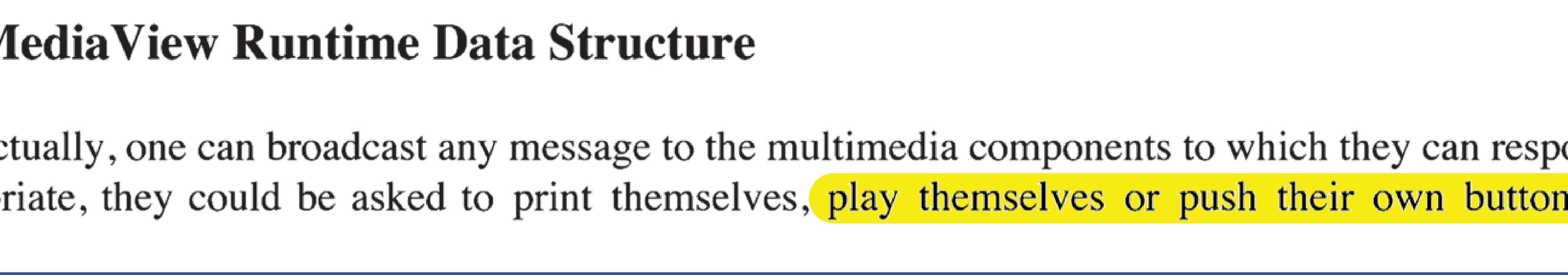
athematica[6] comes bundled with Next computers, it is
ailable. The Mach operating system on the Next plat-
vs applications to communicate easily through pro-
le ports, which is how Media View talks to the
ica application. Alternately, standard Unix pipes
rect communication with the Mathematica server,
its front end. Media View uses both approaches.

1991 IEEE Article: JDX 005



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

omatically invoking the executable application, in response t
tifying of the embed text format, to execute on the client worksta
er to display the object and enable an end-user to directly interact w
ct while the object is being displayed within a display area created
location within the portion of the hypermedia document being displa
rowser-controlled window.



lediaView Runtime Data Structure

ctually, one can broadcast any message to the multimedia components to which they can respo
riate, they could be asked to print themselves, play themselves or push their own button

1991 Usenix Article:

omatically invoking the executable application, in response

tifying of the embed text format, to execute on the client worksta

er to display the object and enable an end-user to directly interact w

ct while the object is being displayed within a display area created

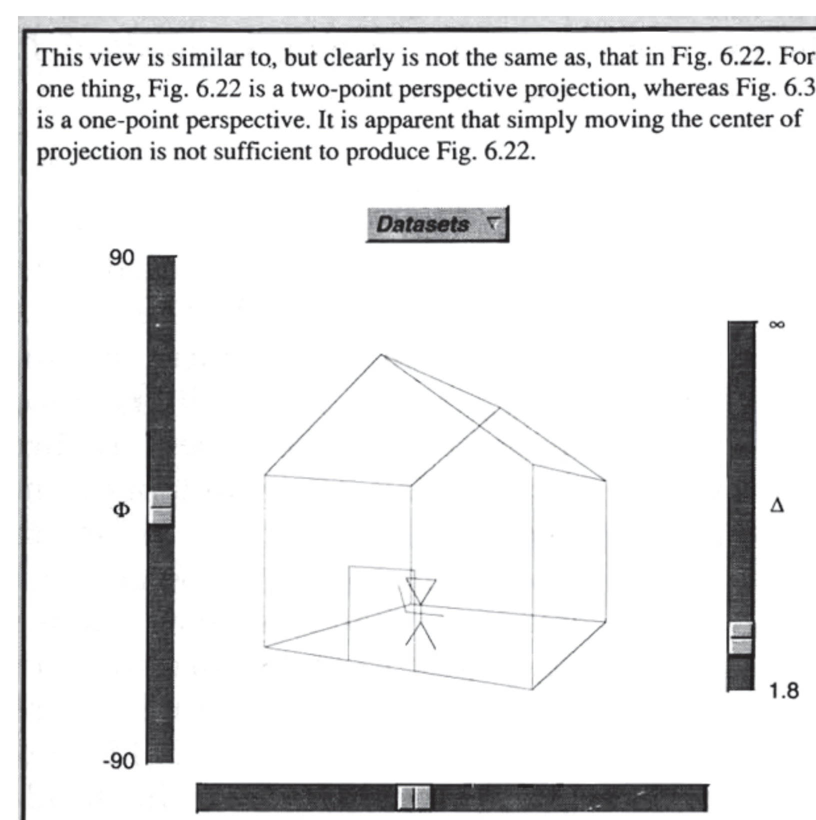
location within the portion of the hypermedia document being displa

browser-controlled window.

Custom Components

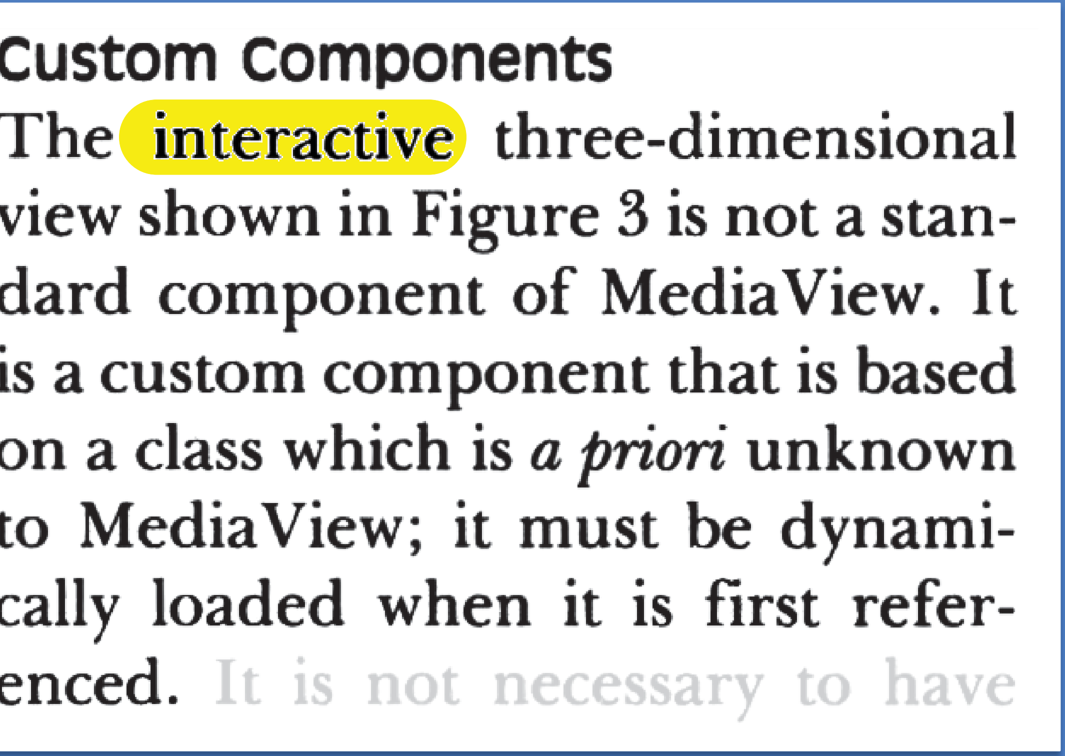
The interactive three-dimensional view shown in Figure 3 is not a standard component of MediaView. It is a custom component that is based on a class which is *a priori* unknown to MediaView; it must be dynamically loaded when it is first referenced. It is not necessary to have

1991 ACM Article: JDX 006



This view is similar to, but clearly is not the same as, that in Fig. 6.22. For one thing, Fig. 6.22 is a two-point perspective projection, whereas Fig. 6.3 is a one-point perspective. It is apparent that simply moving the center of projection is not sufficient to produce Fig. 6.22.

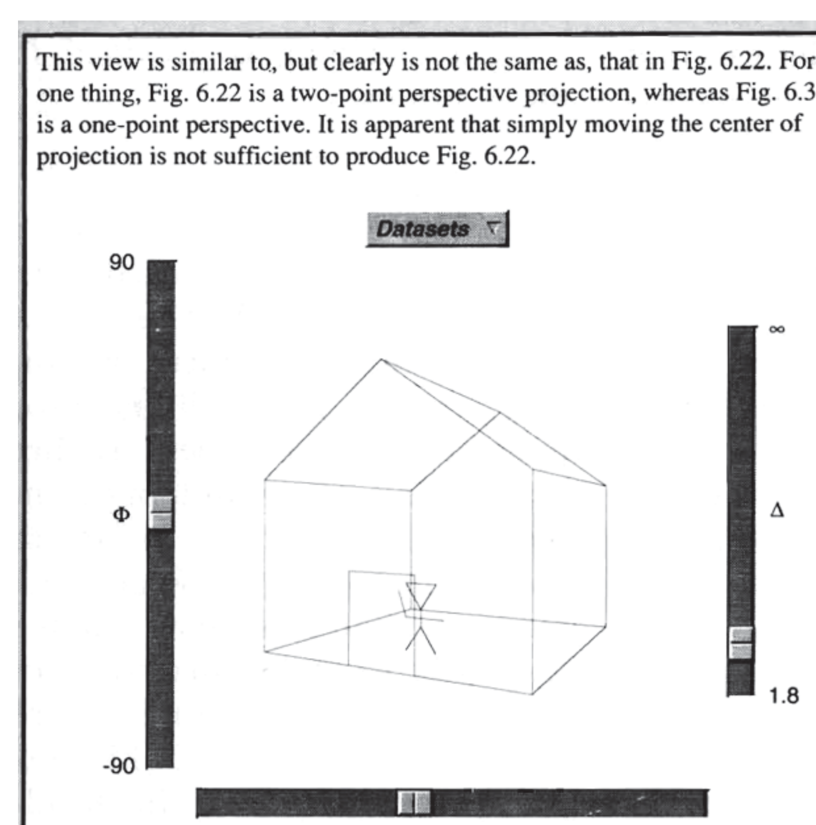
omatically **invoking** the executable application, in response t tifying of the embed text format, to execute on the client worksta er to display the object and **enable an end-user to directly interac object while the object is being displayed within a display** ated at the first location within the portion of the hypermedia doc g displayed in the browser-controlled window.



Custom Components

The interactive three-dimensional view shown in Figure 3 is not a standard component of MediaView. It is a custom component that is based on a class which is *a priori* unknown to MediaView; it must be dynamically loaded when it is first referenced. It is not necessary to have

1991 ACM Article: JDX 006



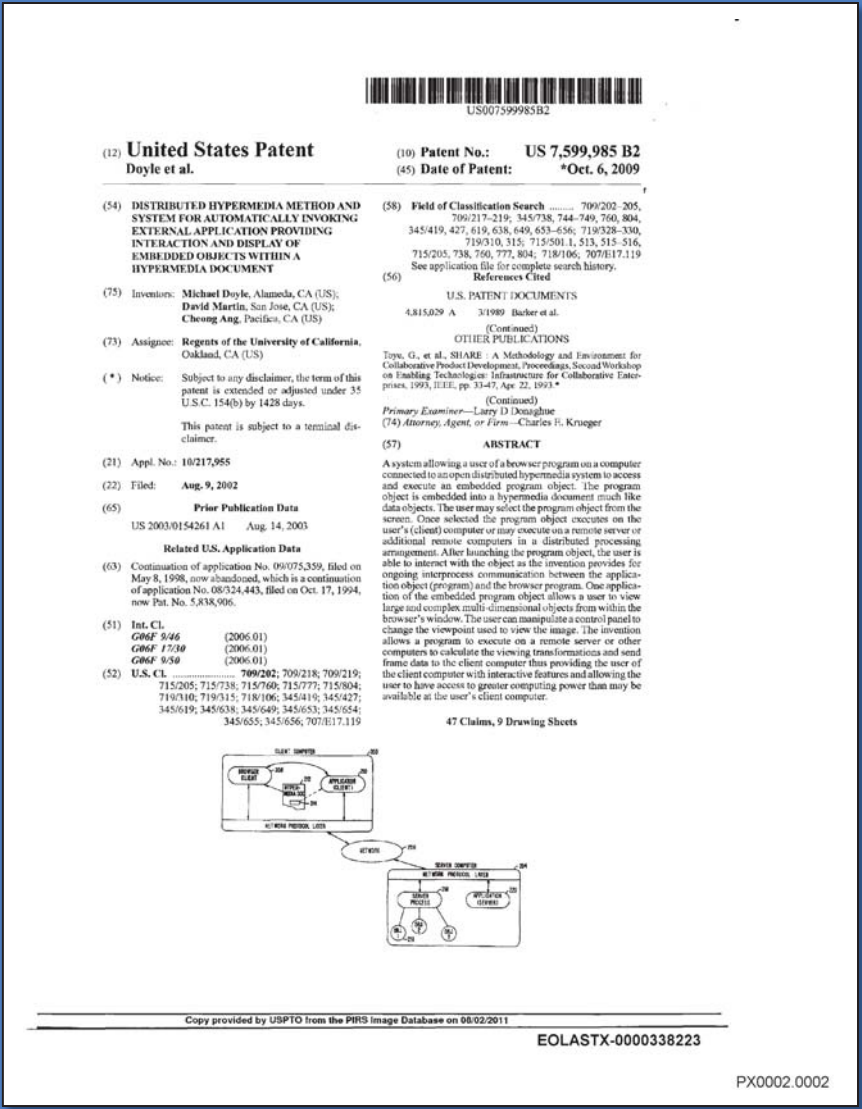
This view is similar to, but clearly is not the same as, that in Fig. 6.22. For one thing, Fig. 6.22 is a two-point perspective projection, whereas Fig. 6.3 is a one-point perspective. It is apparent that simply moving the center of projection is not sufficient to produce Fig. 6.22.

omatically invoking the executable application, in response t tifying of the embed text format, to execute on the client worksta er to display the object and **enable an end-user to directly interac object while the object is being displayed within a display** ated at the first location within the portion of the hypermedia doc g displayed in the browser-controlled window.

’985 Patent Claim 1

ng, at the client workstation from the network server over the network environment, at least one file taining information to enable browser application to display at least portion of distributed hypermedia ument within browser-controlled window;	✓
ing the browser application on the client workstation, with the browser application:	✓
ding to text formats to initiate processing specified by the text formats;	✓
ing at least portion of the document within the browser-controlled window;	✓
ing an embed text format which corresponds to first location in the document, where the embed text hat specifies the location of at least portion of an object external to the file, where the object has type rmation associated with it;	✓
g the type information to identify and locate an executable application external to the file; and	✓
atically invoking the executable application, in response to the identifying of the embed text format, to cute on the client workstation in order to display the object and enable an end-user to directly interact the object while the object is being displayed within a display area created at the first location within the	✓

e evidence that satisfies ’985 Patent Claim 1 also satisfies r asserted claims:



US007599985B2

(12) **United States Patent**
Doyle et al.

(10) **Patent No.:** **US 7,599,985 B2**
(45) **Date of Patent:** ***Oct. 6, 2009**

(54) **DISTRIBUTED HYPERMEDIA METHOD AND SYSTEM FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT**

(75) Inventors: **Michael Doyle**, Alameda, CA (US); **David Martin**, San Jose, CA (US); **Cheong Ang**, Pacifica, CA (US)

(73) Assignee: **Regents of the University of California**, Oakland, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1428 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/217,955**

(22) Filed: **Aug. 9, 2002**

(65) **Prior Publication Data**

US 2003/0154261 A1 Aug. 14, 2003

Related U.S. Application Data

(63) Continuation of application No. 09/075,359, filed on May 8, 1998, now abandoned, which is a continuation of application No. 08/324,443, filed on Oct. 17, 1994, now Pat. No. 5,838,906.

(51) **Int. Cl.**

G06F 9/46	(2006.01)
G06F 17/30	(2006.01)
G06F 9/50	(2006.01)

(52) **U.S. Cl.** **709/202**; 709/218; 709/219; 715/205; 715/738; 715/760; 715/777; 715/804; 719/310; 719/315; 718/106; 345/419; 345/427; 345/619; 345/638; 345/649; 345/653; 345/654; 345/655; 345/656; 707/E17.119

(58) **Field of Classification Search** 709/202-205, 709/217-219; 345/738, 744-749, 760, 804, 345/419, 427, 619, 638, 649, 653-656; 719/328-330, 719/310, 315; 715/501.1, 513, 515-516, 715/205, 738, 760, 777, 804; 718/106; 707/E17.119
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,815,029 A 3/1989 Barker et al.

(Continued)

OTHER PUBLICATIONS

Toye, G., et al., SHARE : A Methodology and Environment for Collaborative Product Development, Proceedings, Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1993, IEEE, pp. 33-47, Apr. 22, 1993.*

(Continued)

Primary Examiner—Larry D Donaghue
(74) *Attorney, Agent, or Firm*—Charles F. Krueger

(57) **ABSTRACT**

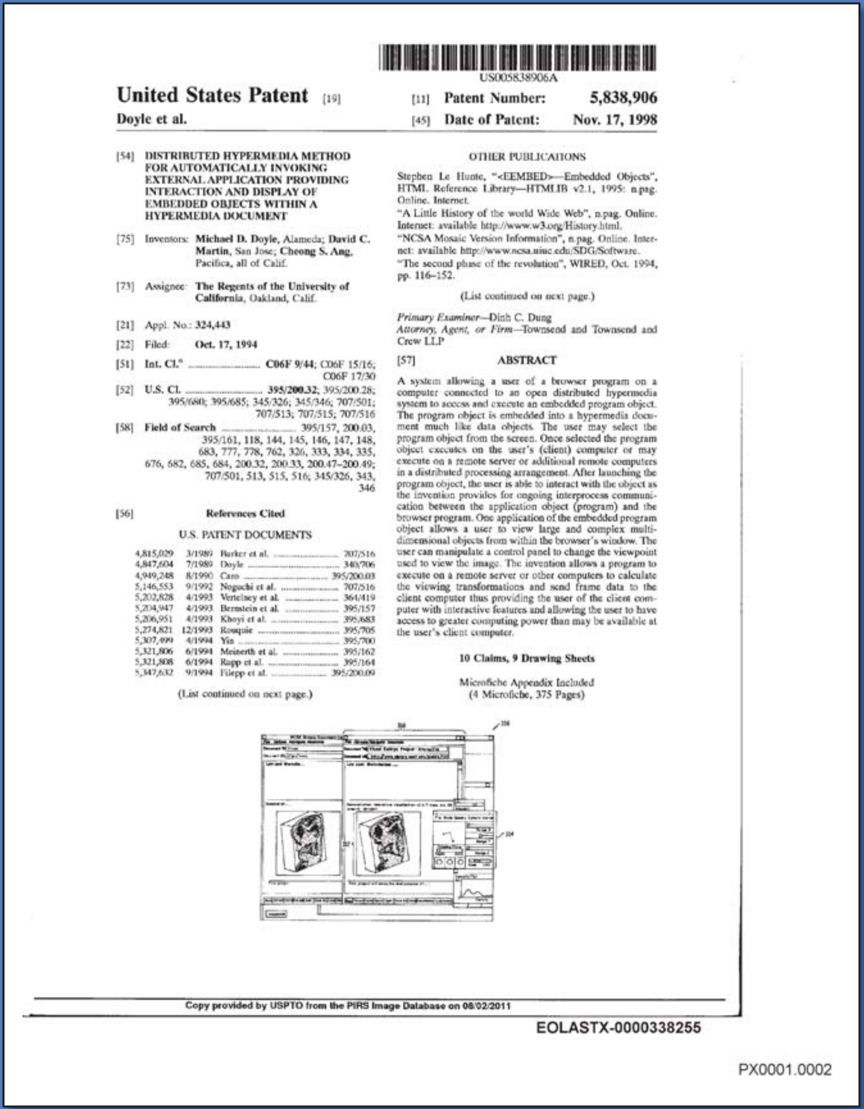
A system allowing a user of a browser program on a computer connected to an open distributed hypermedia system to access and execute an embedded program object. The program object is embedded into a hypermedia document much like data objects. The user may select the program object from the screen. Once selected the program object executes on the user's (client) computer or may execute on a remote server or additional remote computers in a distributed processing arrangement. After launching the program object, the user is able to interact with the object as the invention provides for ongoing interprocess communication between the application object (program) and the browser program. One application of the embedded program object allows a user to view large and complex multi-dimensional objects from within the browser's window. The user can manipulate a control panel to change the viewpoint used to view the image. The invention allows a program to execute on a remote server or other computers to calculate the viewing transformations and send frame data to the client computer thus providing the user of the client computer with interactive features and allowing the user to have access to greater computing power than may be available at the user's client computer.

47 Claims, 9 Drawing Sheets

Copy provided by USPTO from the PIRS Image Database on 08/02/2011

EOLASTX-0000338223

PX0002.0002



US005838906A

United States Patent [19]
Doyle et al.

[11] **Patent Number:** **5,838,906**
[45] **Date of Patent:** **Nov. 17, 1998**

[54] **DISTRIBUTED HYPERMEDIA METHOD FOR AUTOMATICALLY INVOKING EXTERNAL APPLICATION PROVIDING INTERACTION AND DISPLAY OF EMBEDDED OBJECTS WITHIN A HYPERMEDIA DOCUMENT**

[75] Inventors: **Michael D. Doyle**, Alameda; **David C. Martin**, San Jose; **Cheong S. Ang**, Pacifica, all of Calif.

[73] Assignee: **The Regents of the University of California**, Oakland, Calif.

[21] Appl. No.: **324,443**

[22] Filed: **Oct. 17, 1994**

[51] **Int. Cl.⁶** **C06F 9/44**; C06F 15/16; C06F 17/30

[52] **U.S. Cl.** **395/200.32**; 395/200.28; 395/680; 395/685; 345/326; 345/346; 707/501; 707/513; 707/515; 707/516

[58] **Field of Search** 395/157, 200.03, 395/161, 118, 144, 145, 146, 147, 148, 683, 777, 778, 762, 326, 333, 334, 335, 676, 682, 685, 684, 200.32, 200.33, 200.47–200.49; 707/501, 513, 515, 516; 345/326, 343, 346

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,815,029	3/1989	Barker et al.	707/516
4,847,604	7/1989	Doyle	340/706
4,949,248	8/1990	Caro	395/200.03
5,146,553	9/1992	Noguchi et al.	707/516
5,202,828	4/1993	Vertelney et al.	364/419
5,204,947	4/1993	Bernstein et al.	395/157
5,206,951	4/1993	Khoyi et al.	395/683
5,274,821	12/1993	Rouquie	395/705
5,307,499	4/1994	Yin	395/700
5,321,806	6/1994	Meinerth et al.	395/162
5,321,808	6/1994	Rupp et al.	395/164
5,347,632	9/1994	Filepp et al.	395/200.09

(List continued on next page.)

OTHER PUBLICATIONS

Stephen Le Hunte, "<EEMBED>—Embedded Objects", HTML Reference Library—HTMLIB v2.1, 1995: n.pag. Online. Internet.
"A Little History of the world Wide Web", n.pag. Online. Internet: available http://www.w3.org/History.html.
"NCSA Mosaic Version Information", n.pag. Online. Internet: available http://www.ncsa.uiuc.edu/SDG/Software.
"The second phase of the revolution", WIRED, Oct. 1994, pp. 116–152.

(List continued on next page.)

Primary Examiner—Dinh C. Dung
Attorney, Agent, or Firm—Townsend and Townsend and Crew LLP

[57] **ABSTRACT**

A system allowing a user of a browser program on a computer connected to an open distributed hypermedia system to access and execute an embedded program object. The program object is embedded into a hypermedia document much like data objects. The user may select the program object from the screen. Once selected the program object executes on the user's (client) computer or may execute on a remote server or additional remote computers in a distributed processing arrangement. After launching the program object, the user is able to interact with the object as the invention provides for ongoing interprocess communication between the application object (program) and the browser program. One application of the embedded program object allows a user to view large and complex multi-dimensional objects from within the browser's window. The user can manipulate a control panel to change the viewpoint used to view the image. The invention allows a program to execute on a remote server or other computers to calculate the viewing transformations and send frame data to the client computer thus providing the user of the client computer with interactive features and allowing the user to have access to greater computing power than may be available at the user's client computer.

10 Claims, 9 Drawing Sheets

Microfiche Appendix Included
(4 Microfiche, 375 Pages)

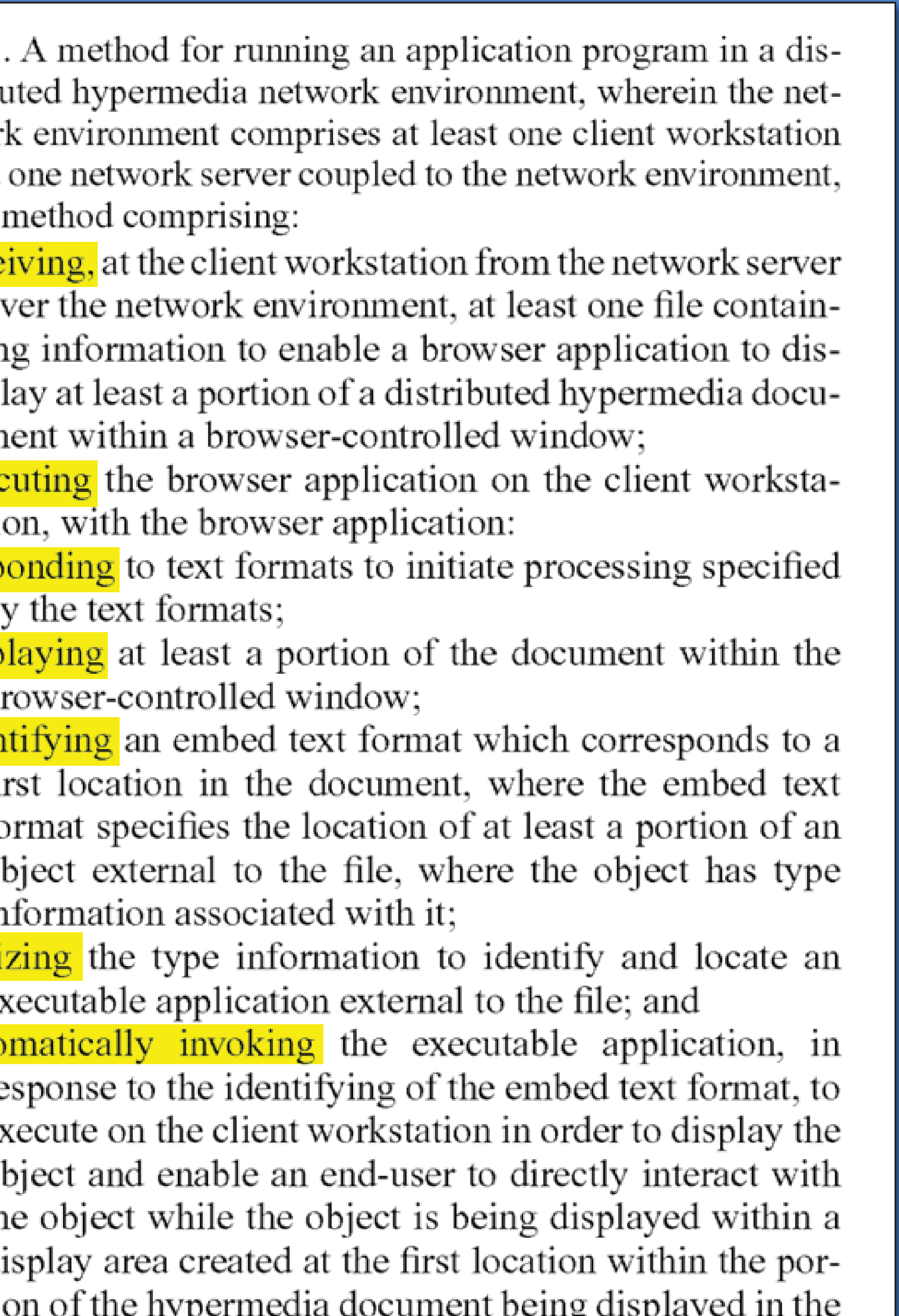
Copy provided by USPTO from the PIRS Image Database on 08/02/2011

EOLASTX-0000338255

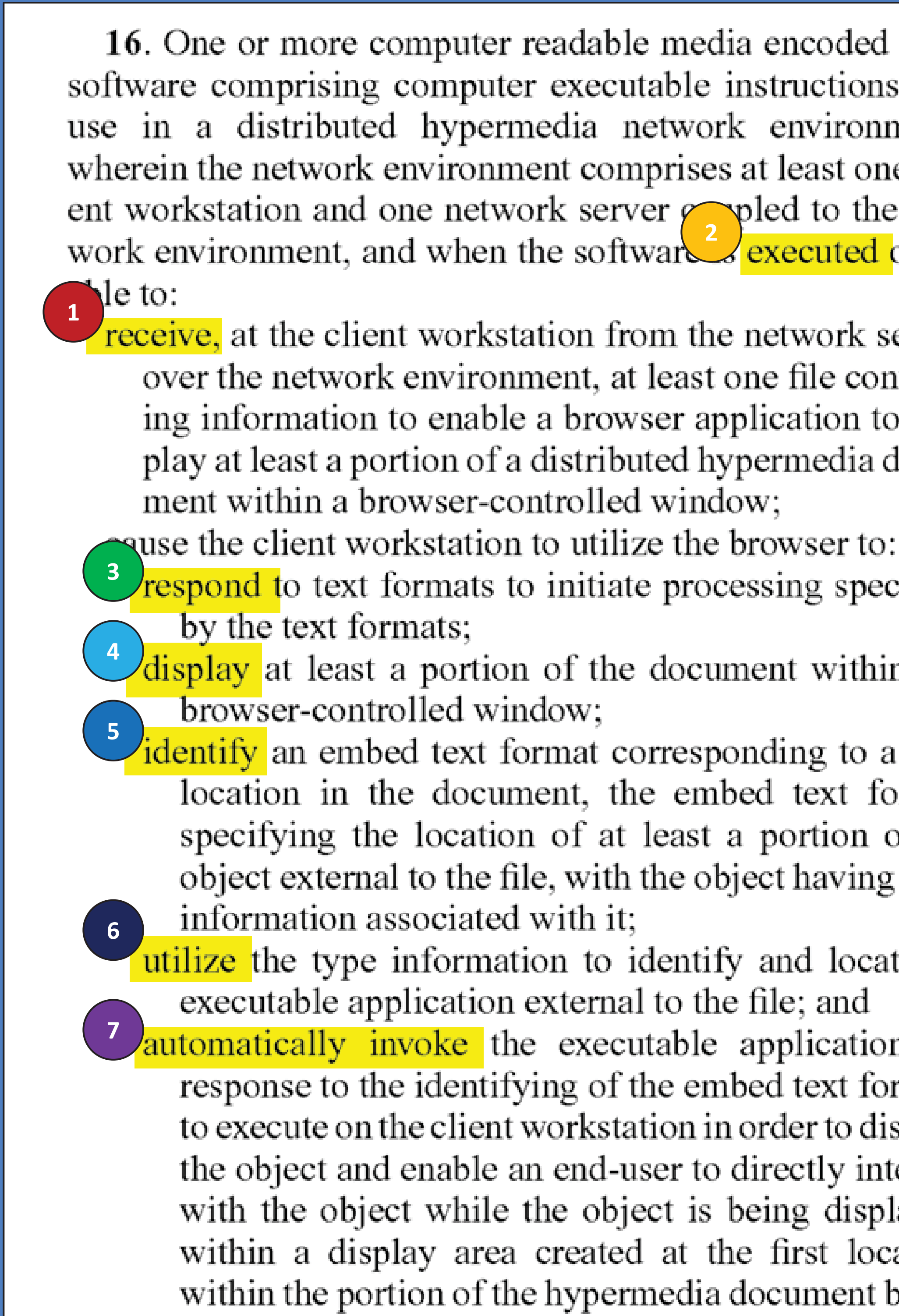
PX0001.0002

’985 Patent Claims: 1, 3, 10, 16, 18, 20, 22, 36, 38, 40, 42

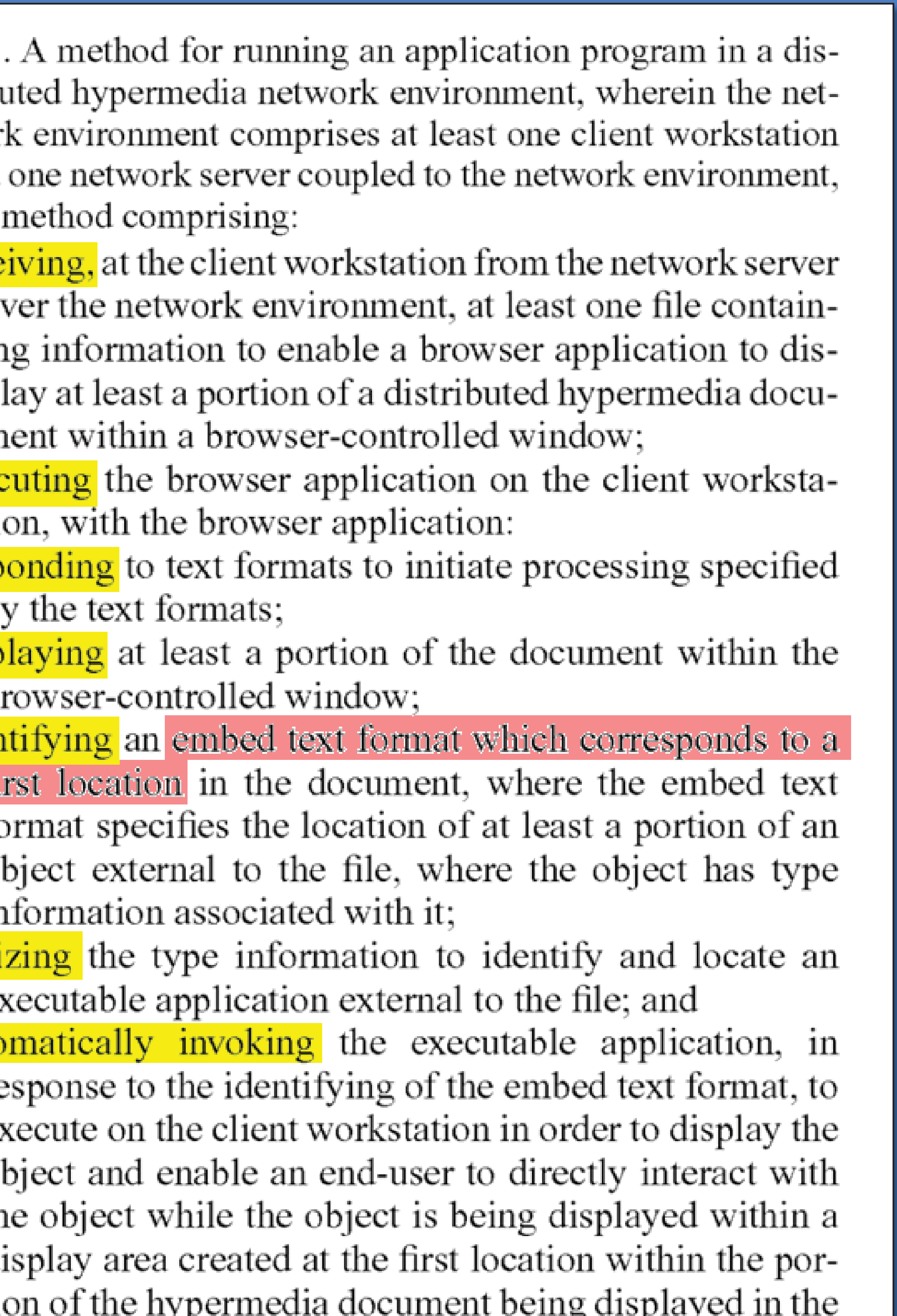
’906 Patent Claims: 1, 6



. A method for running an application program in a dis-
uted hypermedia network environment, wherein the net-
k environment comprises at least one client workstation
one network server coupled to the network environment,
method comprising:
eiving, at the client workstation from the network server
ver the network environment, at least one file contain-
ng information to enable a browser application to dis-
lay at least a portion of a distributed hypermedia docu-
ent within a browser-controlled window;
cuting the browser application on the client worksta-
on, with the browser application:
onding to text formats to initiate processing specified
y the text formats;
laying at least a portion of the document within the
rowser-controlled window;
ntifying an embed text format which corresponds to a
rst location in the document, where the embed text
ormat specifies the location of at least a portion of an
bject external to the file, where the object has type
nformation associated with it;
izing the type information to identify and locate an
xecutable application external to the file; and
omatically invoking the executable application, in
esponse to the identifying of the embed text format, to
xecute on the client workstation in order to display the
bject and enable an end-user to directly interact with
e object while the object is being displayed within a
isplay area created at the first location within the por-
on of the hypermedia document being displayed in the



16. One or more computer readable media encoded
software comprising computer executable instructions
use in a distributed hypermedia network environn
wherein the network environment comprises at least one
ent workstation and one network server coupled to the
work environment, and when the software is executed c
ble to:
receive, at the client workstation from the network se
over the network environment, at least one file con
ing information to enable a browser application to
play at least a portion of a distributed hypermedia d
ment within a browser-controlled window;
cause the client workstation to utilize the browser to:
respond to text formats to initiate processing spec
by the text formats;
display at least a portion of the document withir
browser-controlled window;
identify an embed text format corresponding to a
location in the document, the embed text fo
specifying the location of at least a portion o
object external to the file, with the object having
information associated with it;
utilize the type information to identify and locat
executable application external to the file; and
automatically invoke the executable application
response to the identifying of the embed text for
to execute on the client workstation in order to dis
the object and enable an end-user to directly inte
with the object while the object is being displa
within a display area created at the first loca
within the portion of the hypermedia document b



. A method for running an application program in a dis-
uted hypermedia network environment, wherein the net-
k environment comprises at least one client workstation
one network server coupled to the network environment,
method comprising:

eiving, at the client workstation from the network server
ver the network environment, at least one file contain-
ng information to enable a browser application to dis-
lay at least a portion of a distributed hypermedia docu-
ent within a browser-controlled window;

cuting the browser application on the client worksta-
on, with the browser application:

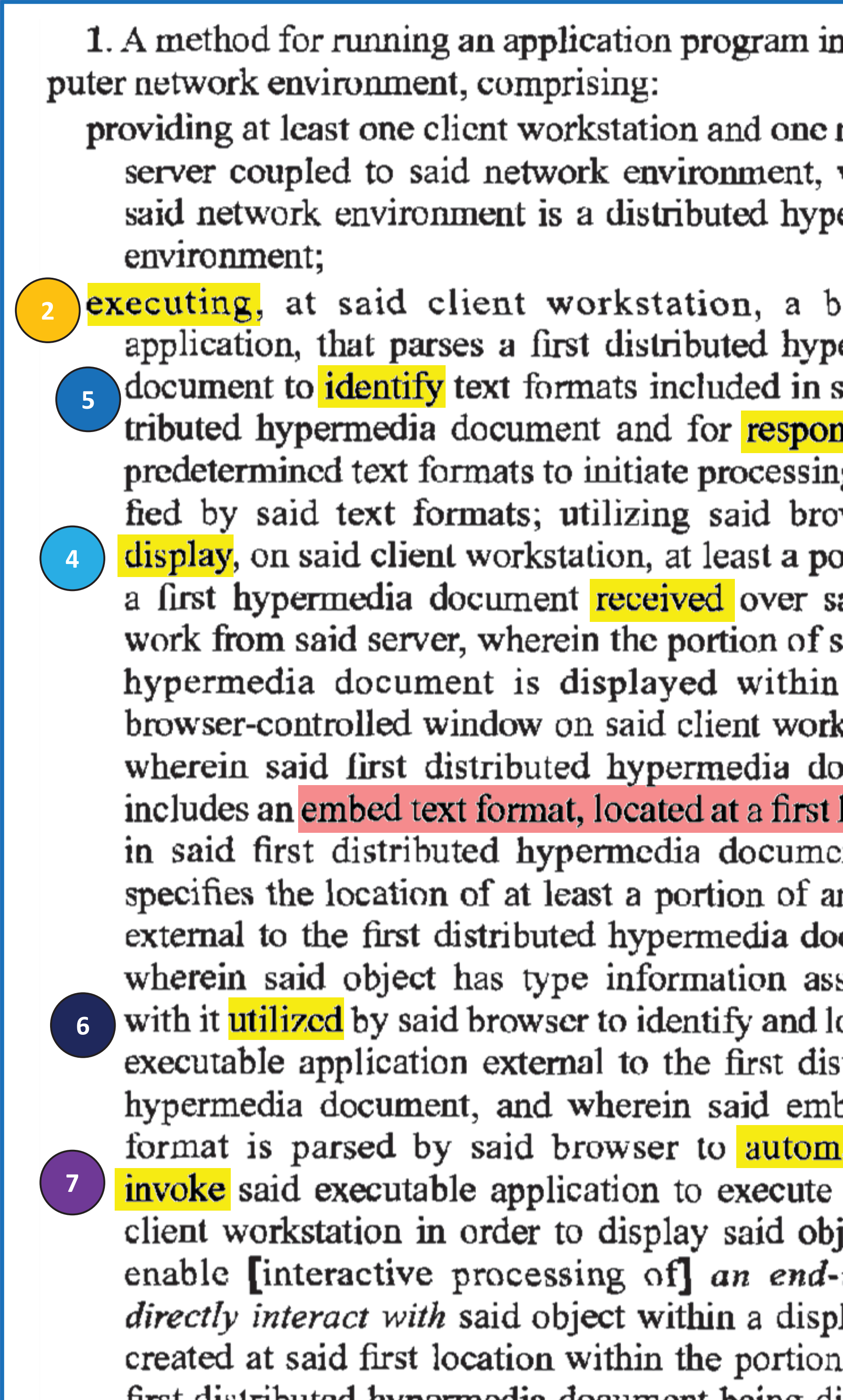
onding to text formats to initiate processing specified
y the text formats;

laying at least a portion of the document within the
rowser-controlled window;

tifying an embed text format which corresponds to a
rst location in the document, where the embed text
ormat specifies the location of at least a portion of an
bject external to the file, where the object has type
nformation associated with it;

izing the type information to identify and locate an
xecutable application external to the file; and

omatically invoking the executable application, in
esponse to the identifying of the embed text format, to
xecute on the client workstation in order to display the
bject and enable an end-user to directly interact with
ne object while the object is being displayed within a
isplay area created at the first location within the por-
on of the hypermedia document being displayed in the



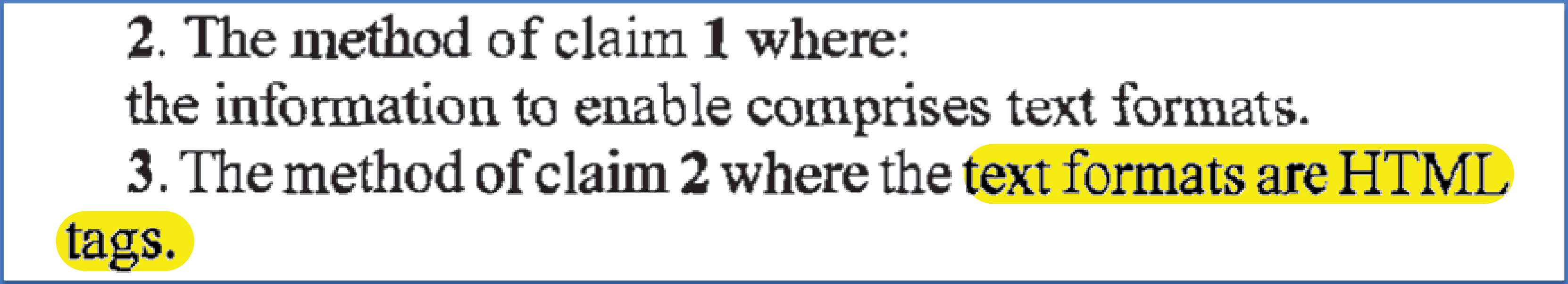
1. A method for running an application program in
puter network environment, comprising:

providing at least one client workstation and one
server coupled to said network environment,
said network environment is a distributed hype
environment;

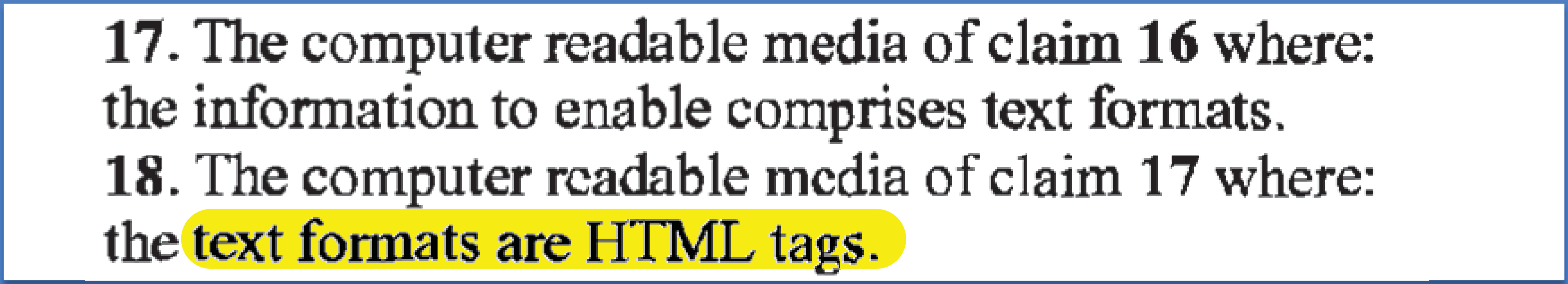
executing, at said client workstation, a b
application, that parses a first distributed hype
document to identify text formats included in s
tributed hypermedia document and for respon
predetermined text formats to initiate processing
fied by said text formats; utilizing said bro
display, on said client workstation, at least a po
a first hypermedia document received over sa
work from said server, wherein the portion of s
hypermedia document is displayed within
browser-controlled window on said client work
wherein said first distributed hypermedia do
includes an embed text format, located at a first l
in said first distributed hypermedia docume
specifies the location of at least a portion of an
external to the first distributed hypermedia doc
wherein said object has type information ass
with it utilized by said browser to identify and lo
executable application external to the first dis
hypermedia document, and wherein said emb
format is parsed by said browser to autom
invoke said executable application to execute
client workstation in order to display said obj
enable [interactive processing of] *an end-*
directly interact with said object within a displ
created at said first location within the portion
first distributed hypermedia document being di

’985 Patent

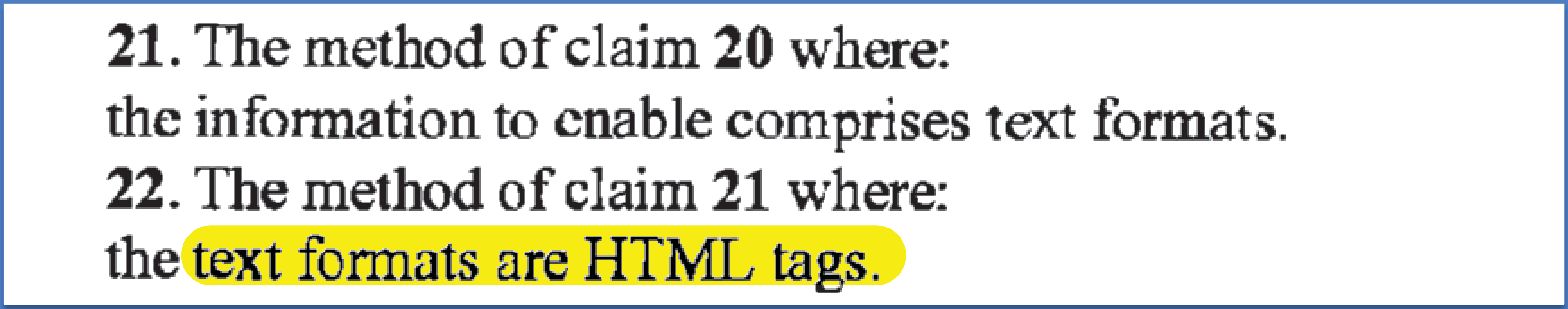
	Claim 1	Claim 3	Claim 10	Claim 16	Claim 18	Claim 20	Claim 22	Claim 36	Claim 38	Claim 40	Claim 42
nning an application program in rmedia network environment, work environment comprises at workstation and one network o the network environment, the ing:	✓	✓	✓	Computer Readable Media	Computer Readable Media	Serving	✓	✓	✓	Serving	Serving
client workstation from the ver the network environment, at taining information to enable ion to display at least portion of media document within browser-w;	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
owser application on the client the browser application:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
xt formats to initiate processing text formats;	✓	✓	✓	✓	HTML Tags	✓	HTML Tags	✓	HTML Tags	✓	HTML Tags
st portion of the document within trolled window;	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bed text format which rst location in the document, d text format specifies the st portion of an object external to e object has type information t;	✓	✓	Type information in embed text format	✓	✓	✓	✓	✓	✓	✓	✓
information to identify and locate pplication external to the file; and	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
king the executable application, in entifying of the embed text format, to ent workstation in order to display the an end-user to directly interact with	✓	HTML	✓	✓	✓	✓	✓	Distributed	Distributed	Distributed	Distributed Applicatio



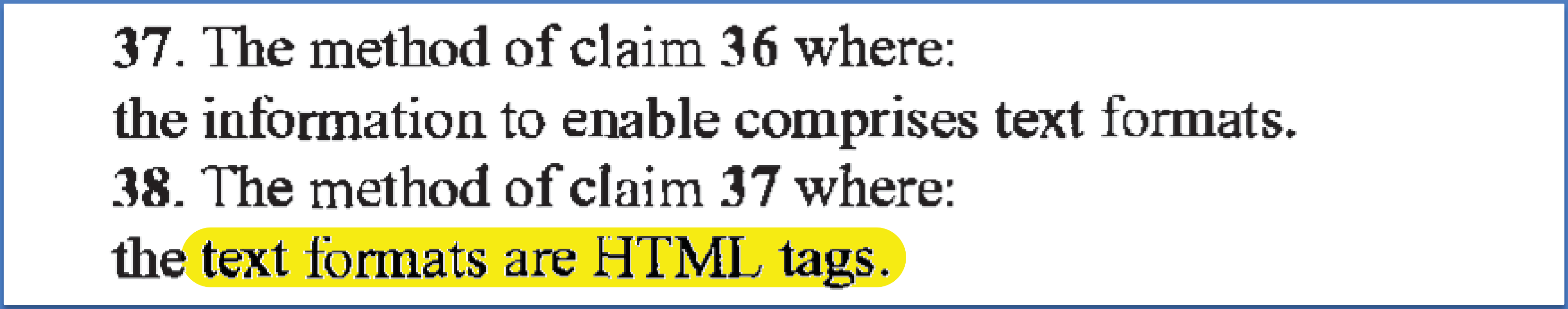
2. The method of claim **1** where:
the information to enable comprises text formats.
3. The method of claim **2** where the text formats are HTML tags.



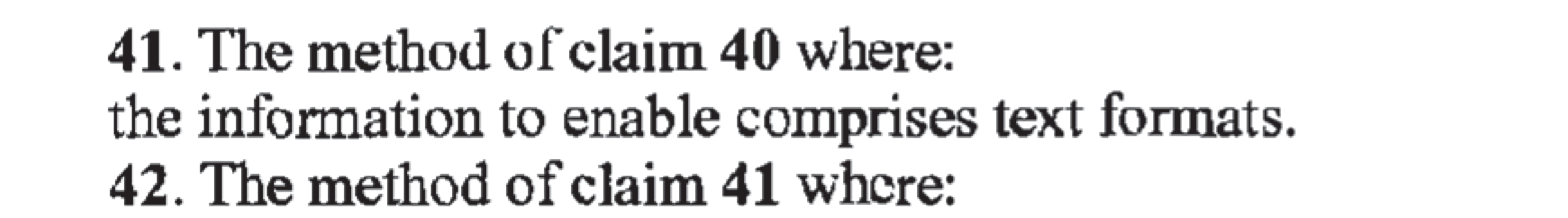
17. The computer readable media of claim **16** where:
the information to enable comprises text formats.
18. The computer readable media of claim **17** where:
the text formats are HTML tags.



21. The method of claim **20** where:
the information to enable comprises text formats.
22. The method of claim **21** where:
the text formats are HTML tags.

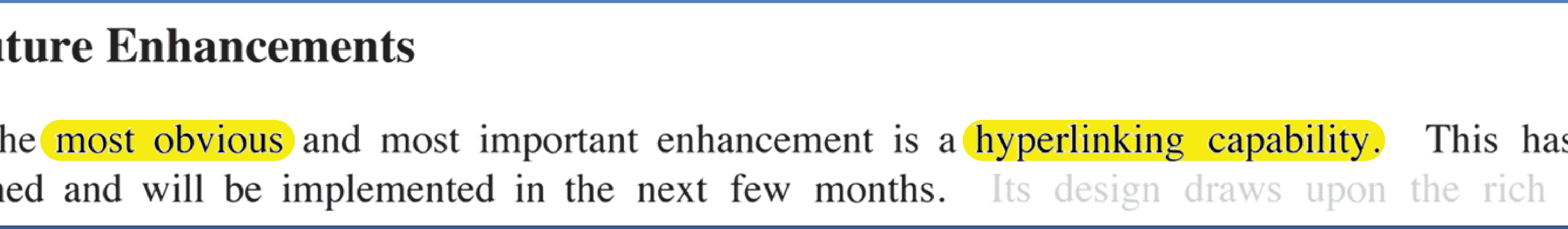


37. The method of claim **36** where:
the information to enable comprises text formats.
38. The method of claim **37** where:
the text formats are HTML tags.



41. The method of claim **40** where:
the information to enable comprises text formats.
42. The method of claim **41** where:

oted in 1991 that including hyperlinking in ediaView was obvious:

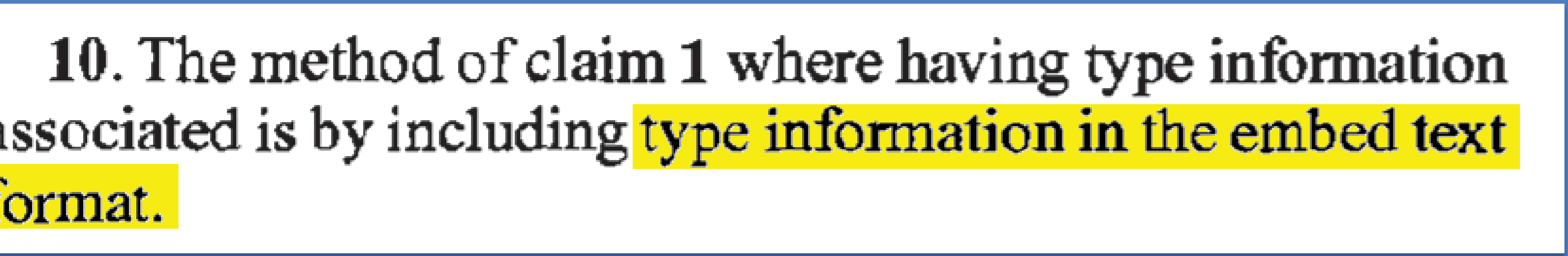


ture Enhancements

he most obvious and most important enhancement is a hyperlinking capability. This has ned and will be implemented in the next few months. Its design draws upon the rich

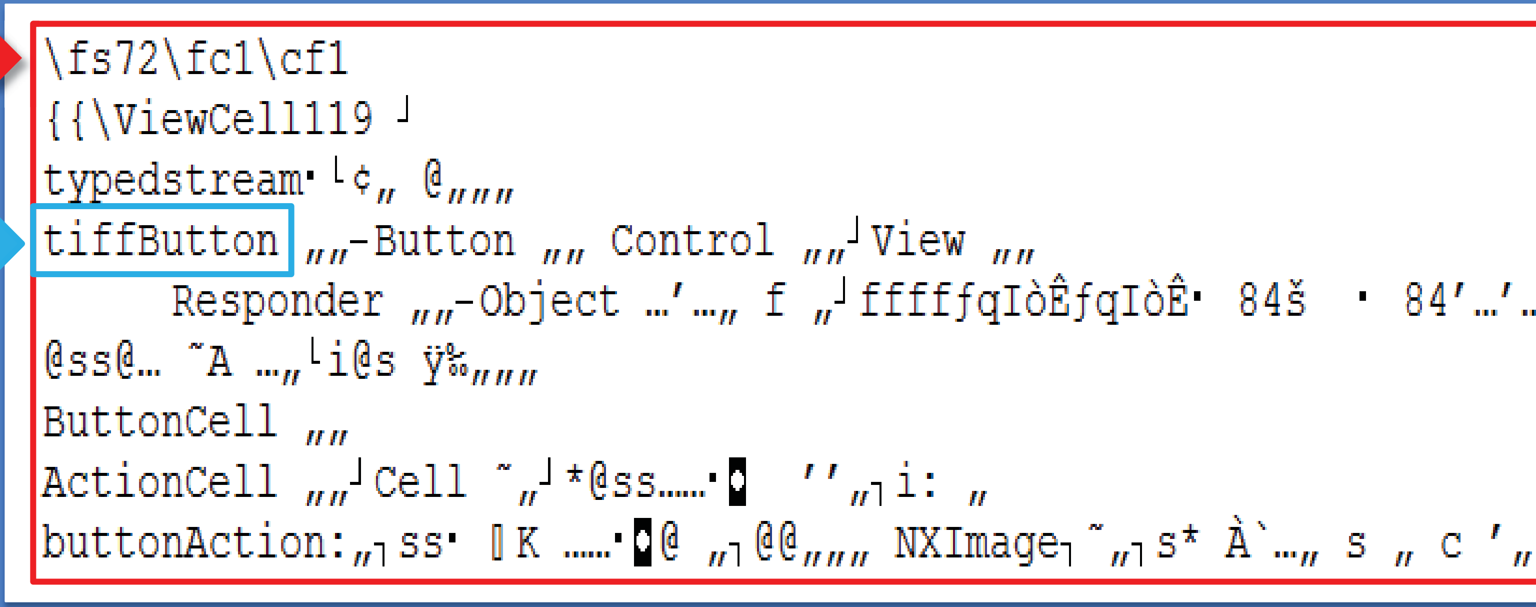
1991 Usenix Article: JI

bvious to combine MediaView with other browsers, e CERN browser



10. The method of claim 1 where having type information
ssociated is by including type information in the embed text
ormat.

ediaView Hypermedia File Excerpt:



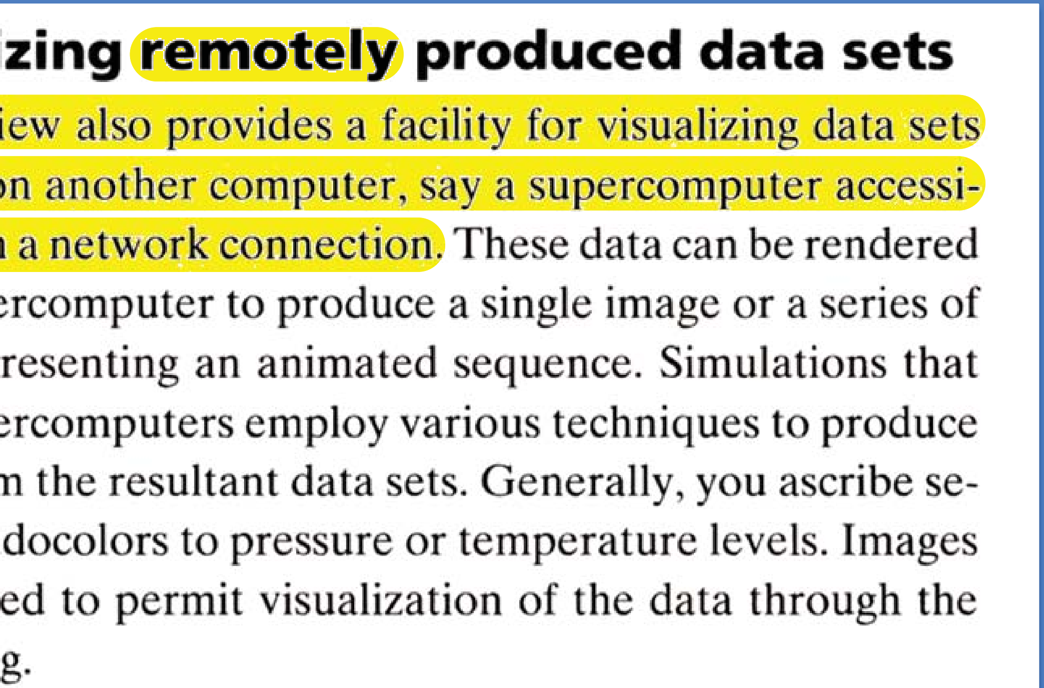
```
\fs72\fc1\cf1
{{\ViewCell119 ┘
typedstream·└¢„ @„„„
tiffButton „„-Button „„ Control „„┘View „„
     Responder „„-Object …′…„ f „┘ffffqIòÊfqIòÊ· 84š  · 84′…′…
@ss@… ˜A …„└i@s ÿ‰„„„
ButtonCell „„
ActionCell „„┘Cell ˜„┘*@ss……·█  ′′„┐i: „
buttonAction:„┐ss· ▯K ……·█@ „┐@@„„„ NXImage┐˜„┐s* À`…„ s „ c ′„
```

mbed text format

ation associated
thematica object

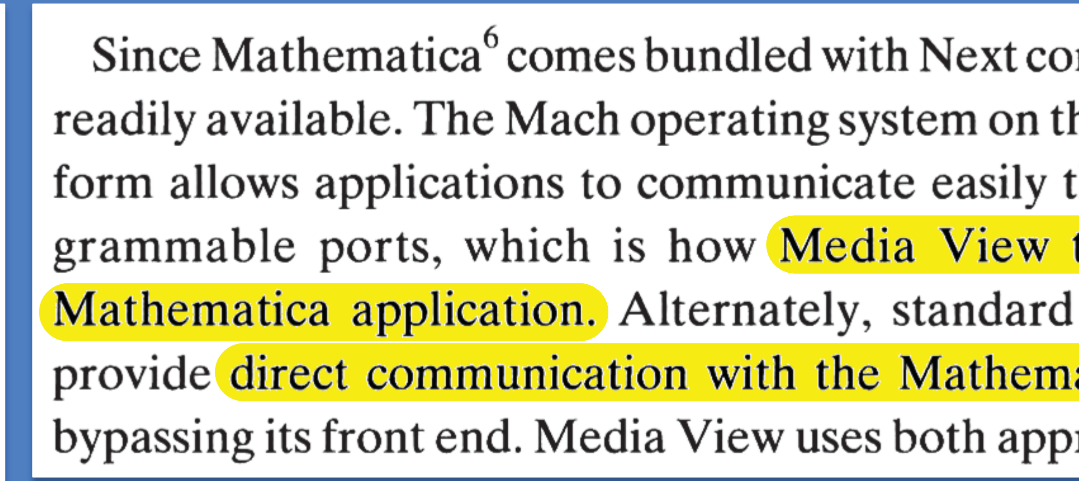
m 36 & 40: … wherein the executable application is part of a tributed application, and wherein at least a portion of the distrib lication is for execution on a remote network server coupled to th ributed hypermedia network environment.

rt on "distributed application": an application that is **capable** of being ken up and performed among two or more computers



izing remotely produced data sets

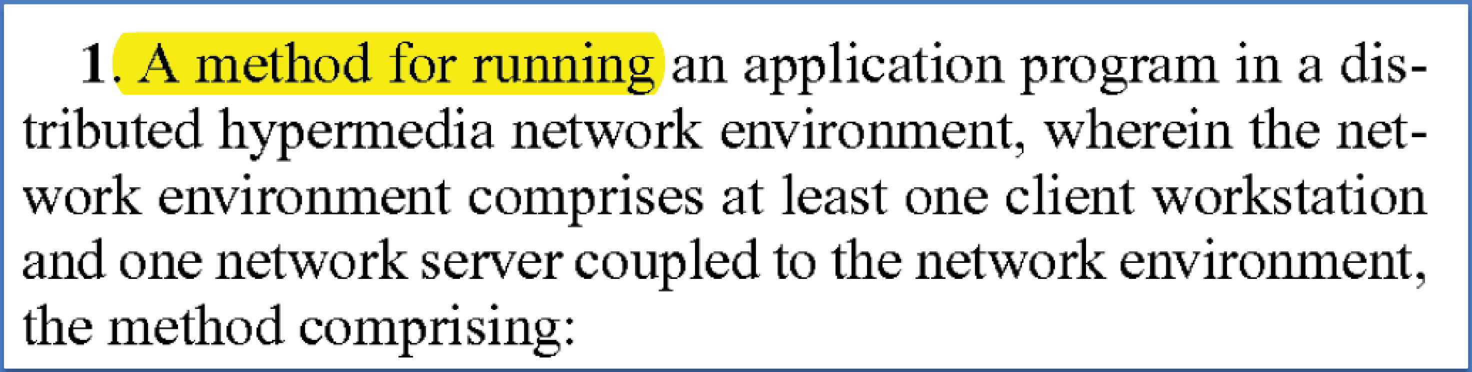
iew also provides a facility for visualizing data sets on another computer, say a supercomputer accessi- a network connection. These data can be rendered ercomputer to produce a single image or a series of resenting an animated sequence. Simulations that ercomputers employ various techniques to produce m the resultant data sets. Generally, you ascribe se- docolors to pressure or temperature levels. Images ed to permit visualization of the data through the g.



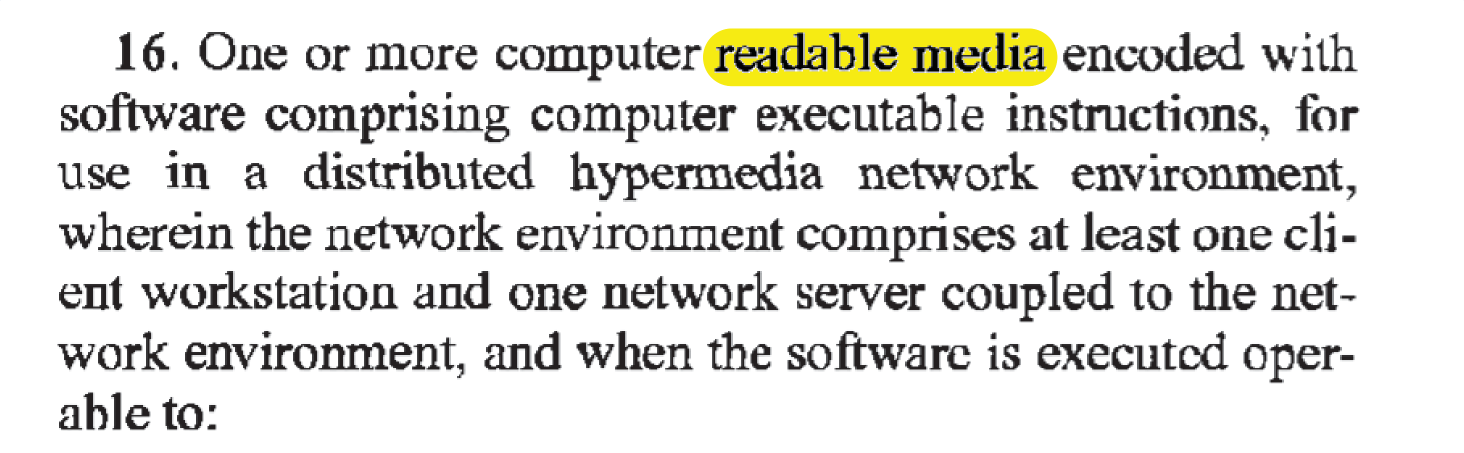
Since Mathematica[6] comes bundled with Next co readily available. The Mach operating system on th form allows applications to communicate easily t grammable ports, which is how Media View t Mathematica application. Alternately, standard provide direct communication with the Mathema bypassing its front end. Media View uses both app

1991 IEEE Article

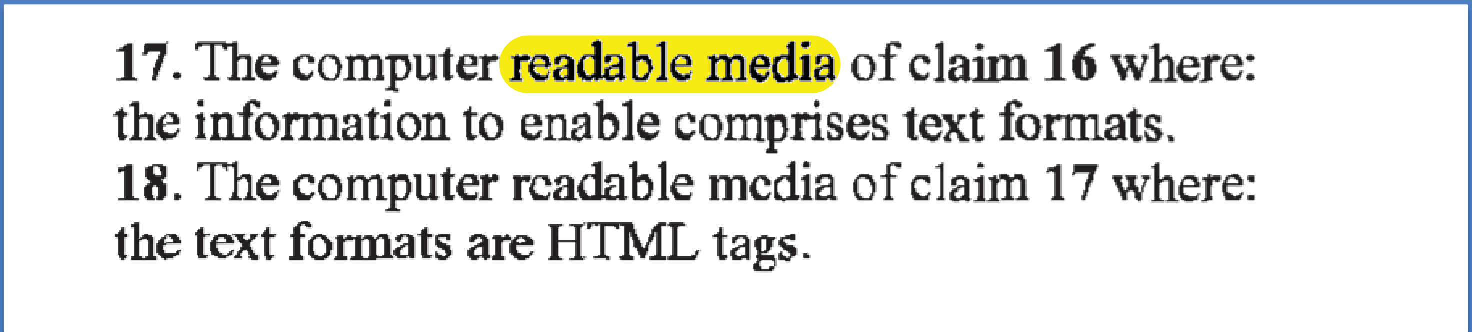
atent



1. A method for running an application program in a distributed hypermedia network environment, wherein the network environment comprises at least one client workstation and one network server coupled to the network environment, the method comprising:



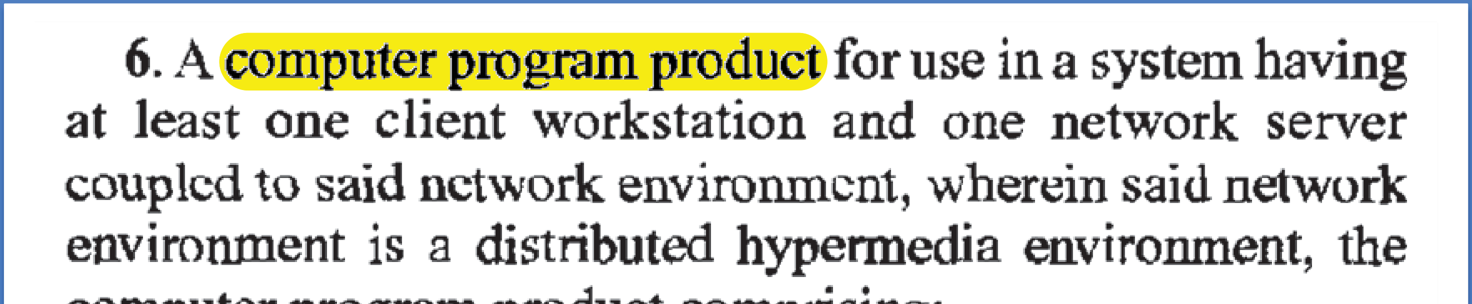
16. One or more computer readable media encoded with software comprising computer executable instructions, for use in a distributed hypermedia network environment, wherein the network environment comprises at least one client workstation and one network server coupled to the network environment, and when the software is executed operable to:



17. The computer readable media of claim 16 where: the information to enable comprises text formats.

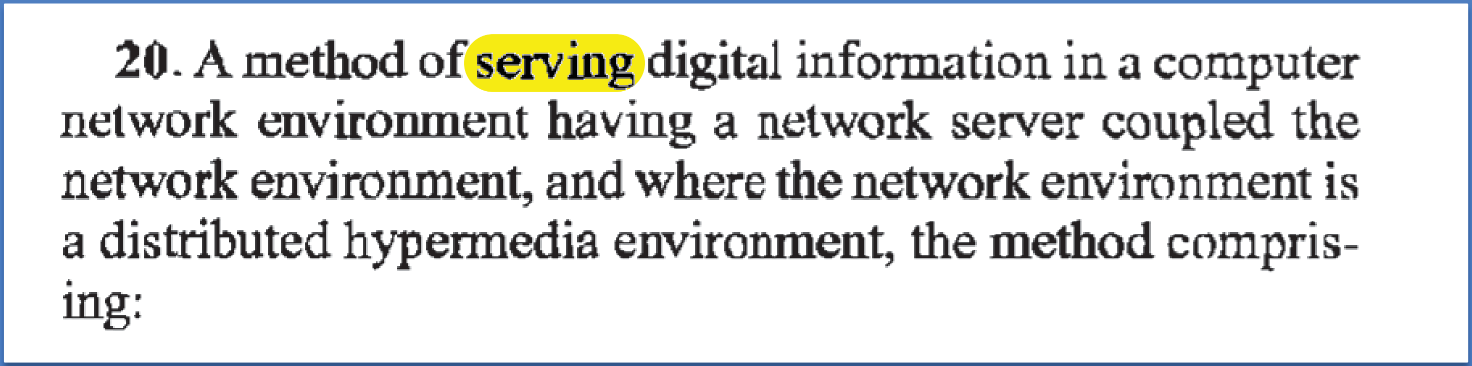
18. The computer readable media of claim 17 where: the text formats are HTML tags.

atent

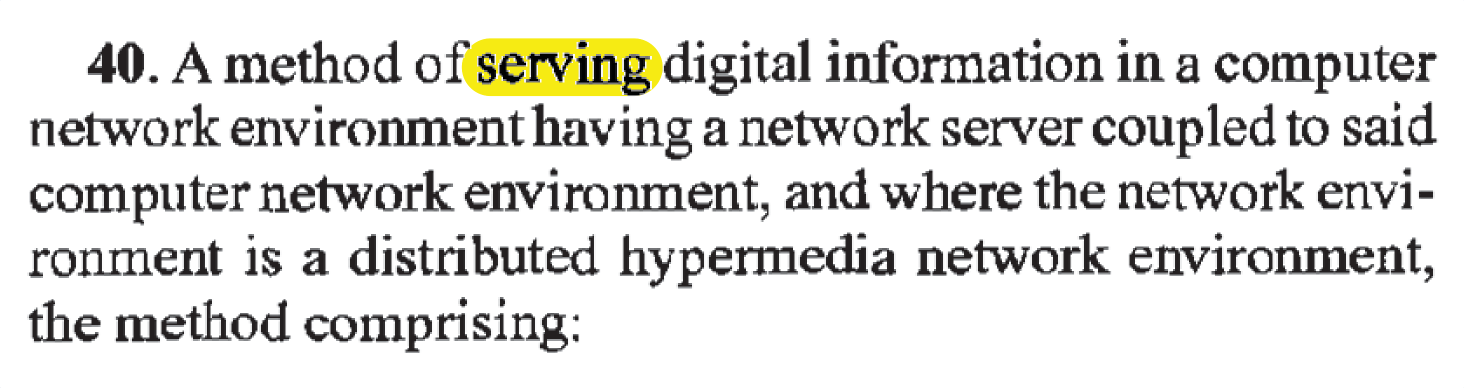


6. A computer program product for use in a system having at least one client workstation and one network server coupled to said network environment, wherein said network environment is a distributed hypermedia environment, the computer program product comprising:

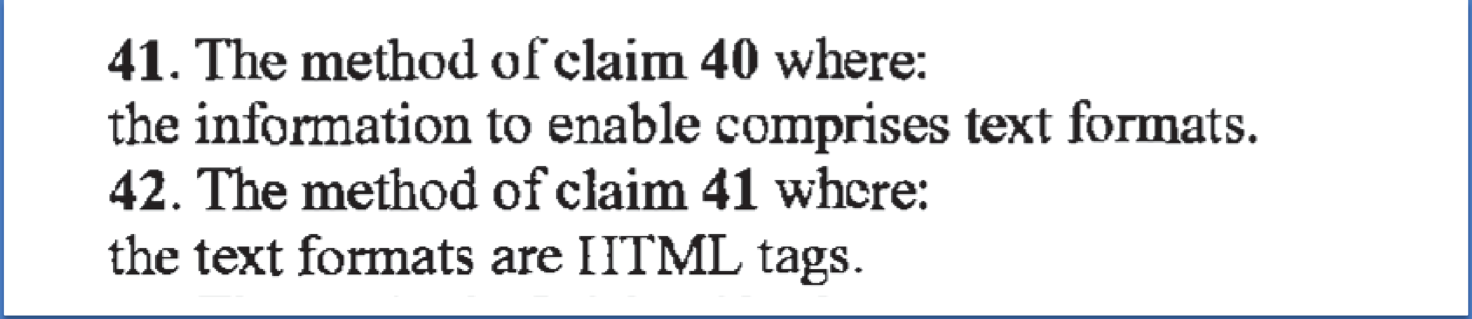
Patent



20. A method of serving digital information in a computer network environment having a network server coupled the network environment, and where the network environment is a distributed hypermedia environment, the method comprising:



40. A method of serving digital information in a computer network environment having a network server coupled to said computer network environment, and where the network environment is a distributed hypermedia network environment, the method comprising:



41. The method of claim **40** where:
the information to enable comprises text formats.
42. The method of claim **41** where:
the text formats are HTML tags.

’985 Patent

	Claim 1	Claim 3	Claim 10	Claim 16	Claim 18	Claim 20	Claim 22	Claim 36	Claim 38	Claim 40	Claim 42
nning an application program in rmedia network environment, work environment comprises at workstation and one network o the network environment, the ing:	✓	✓	✓	Computer Readable Media ✓	Computer Readable Media ✓	Serving ✓	✓	✓	✓	Serving ✓	Serving ✓
client workstation from the ver the network environment, at taining information to enable ion to display at least portion of media document within browser- w;	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
owser application on the client the browser application:	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
xt formats to initiate processing text formats;	✓	✓	✓	✓	HTML Tags ✓	✓	HTML Tags ✓	✓	HTML Tags ✓	✓	HTML Tags ✓
st portion of the document within trolled window;	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bed text format which rst location in the document, d text format specifies the st portion of an object external to e object has type information t;	✓	✓	Type information to embed text format ✓	✓	✓	✓	✓	✓	✓	✓	✓
information to identify and locate pplication external to the file; and	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
king the executable application, in entifying of the embed text format, to ent workstation in order to display the an end-user to directly interact with	✓	HTML ✓	✓	✓	✓	✓	✓	Distributed ✓	Distributed ✓	Distributed ✓	Distributed ✓

asserted claims are anticipated or obvious based on Viola

- Viola disclosed the claims more than one year before the patent was file
- Obvious to combine any of the prior art to fill any gaps

asserted claims are obvious based on MediaView

- Obvious to combine MediaView with CERN web browser