U.S. PROVISIONAL PATENT APPLICATION
FOR:
METHOD AND APPARATUS FOR BROWSING USING MULTIPLE COORDINATED DEVICE SETS

INVENTOR:

RICHARD R. REISMAN

MORGAN & FINNEGAN, LLP
345 PARK AVENUE
NEW YORK, NY 10154-0053
(212) 758-4800
(212) 751-6849 (FACSIMILE)
WWW.MORGANFINNEGAN.COM

TITLE OF THE INVENTION:

Method And Apparatus For Browsing Using Multiple Coordinated Device Sets.

Field of the Invention

The present invention is directed generally to interactive television and similar interactive hypermedia such as from television or Internet sources, and more particularly to the provision and use of user interfaces that permit interaction using of multiple coordinated device sets.

Background of the Invention

While "convergence" of television (TV) and computer technology have been a major focus of innovation and commercial development since the early 1990s, particularly in the area of "interactive television" (ITV), there remains a huge gulf in the nature of the user experience of ITV and of computer-based media such as the World Wide Web. Convergence has taken hold in infrastructure technologies, with digital and computer-based TV (DTV) editing, production, distribution, transmission, and devices. At heart ITV is a matter of hypermedia browsing, the process of browsing linked media resources like the Web, differing only on its emphasis on video as the central medium.

However, there remains a divide relating to the dramatic difference in how TV-centric and computer-centric media are used, and to the cultural divide between the TV production and distribution industry and the computer and Web industries that has prevented a convergence in user experience from developing or even being seen as possible and desirable. TV usage and directions are focused on its character as a lean-back, across-the-room, low resolution, and relatively passive, relaxed experience of couch potatoes viewing large, often shared TV screens

with simple remote controls. PC usage and directions are focused on its character as lean-forward, up-close, high resolution, and intensive, highly interactive experiences of individuals with PC-styles displays, keyboards, and pointing devices. Variant device sets and applications, such as PDAs, tablets, and video games, could be taken as suggestive of the desirability of selecting among alternative usage modes and form factors, but only very limited aspects of these suggestions have been recognized.

The limitations of these radically disparate device set form factors have severely limited the appeal of ITV. ITV promises to greatly enrich the TV experience by allowing interactive features that can range from access to supplementary enhancement material such as background on programs, casts and players, sports statistics, polls, chat messaging, and interactive advertisements and purchase offers ("t-commerce"), and all manner of other tangential information, to ways to vary the core program content by acting on viewer input and choices as to camera angles or even alternative plots, as well as providing improved control of the core experience with electronic program guides (EPGs), personal video recorders (PVRs) and video on demand (VOD) and similar features.

The problem is that these interactive features are not well served by the TV usage mode and form factor, and their use interferes with the basic TV experience. Rich interaction with a TV is inherently difficult. Presentation of information is limited by the poor capabilities of a TV screen for presenting text, menus, and navigations controls, and the crude input capabilities of a remote control. The rich information and navigation functionality available on a Web browser or other PC-based user interface (e.g., UI, especially graphical user interfaces, GUIs) must be "dumbed-down" and limited for use on a TV, and even use of high-definition TV (HDTV) may not significantly ease that—people do not like to read or do fine work from across-the-room, it is

just not comfortable ergonomics. Furthermore, the attempt to show interactive controls and enhancements on the TV interferes with viewing by the person interacting, as well as any other viewers in the room. Compounding these issues and slowing recognition of better solutions is the dominance of the cable TV industry, its struggles in developing and deploying the advanced set-top boxes (STBs) needed to offer meaningful ITV services of the form it envisions, and its orientation to closed, proprietary systems that do not fully exploit or adapt to advances in the PC and Internet world.

The computer community has attempted to market PCs that include a TV tuner to support TV function in a PC-centric model, as promoted by the PC-DTV Consortium. However, these systems suffer from the converse problem, in that their form factors are not suited to the fact that most people do not want to watch TV at a PC, with its lean-forward, up-close form factor. Furthermore, such devices cannot effectively receive protected cable or satellite programming. And here, as with conventional TVs, the use of a single system forces technical, economical, and usage constraints on the inherently complex, multi-tasking, man-machine behavior that is desired in a rich hypermedia browsing experience.

There has also been some recognition that PCs provide a way around the limited installed base of advanced STBs, but this is generally perceived only as a limited stopgap. So called Enhanced TV or Extended TV or "telewebbing" has emerged to exploit the fact that tens of millions of households have PCs in the same room as their TVs, and can surf related content on the Web while watching TV. Some broadcasters such as ABC and PBS have exploited this to offer Web content synchronized to a TV program, but it is the user who must coordinate the use of the PC with the TV, by finding the appropriate Web site. In spite of the fact that the installed base for such open hardware is some ten times that of ITV-capable set-top boxes, the ITV

community generally views such "two-box" solutions as an unfortunate and awkward stopgap that may be desirably supplanted by advanced "one-box" systems whose wide deployment must be awaited. Some major reasons for this lack of acceptance are that this simplistic two-box model supports only very limited, pre-defined synchronization of the availability of TV and enhancement content that is built into a rigidly fixed two-box structure at the content source, and, even more importantly, that it completely fails to address any coordination of user activity at the two separate boxes.

Across all of this, the key elements that are lacking are provision of a broadly flexible, powerful, selective, and simple user interface paradigm for browsing hypermedia across multiple device sets, whether they are integrated or not, with related methods for user and/or authoring control of such a UI, and provision of an effective method for independent systems to coordinate browsing activities to enable such a user interface to be employed across multiple independent systems. Further lacking across all of these aspects is delivery of these services in a way that provides the user with a smoothly integrated experience in which interactions on the multiple systems are coupled or decoupled to the degree appropriate to the task of the moment.

Summary of Various Embodiments the Invention

According to embodiments of the present invention there are provided systems and methods for navigating hypermedia using multiple coordinated input/output device sets. Embodiments of the invention allow a user and/or an author to control what resources are presented on which device sets (whether they are integrated or not), and provide for coordinating browsing activities to enable such a user interface to be employed across multiple independent systems.

Brief Description of the Drawings

Further aspects of the instant invention will be more readily appreciated upon review of the detailed description of the preferred embodiments included below when taken in conjunction with the accompanying drawings, of which:

FIG. 1 is a block diagram of an exemplary assemblage of user systems, networks, and remote services for implementing certain embodiments of the present invention.

FIG. 2 is a set of block diagrams of exemplary groupings of device sets and systems in the assemblage of FIG. 1.

FIG. 3 is a schematic diagram of a number of exemplary user interface display layouts according to certain embodiments of the present invention.

FIG. 4 is a schematic diagram of an exemplary structure for state information relating to systems within the assemblage of FIG. 1, relating to the coordination of a multimachine user interface according to certain embodiments of the present invention.

FIG. 5 is a schematic diagram of an exemplary process, performed by the systems of FIG. 1, for transferring state data according to certain embodiments of the present invention.

FIG. 6 is a flow chart of an exemplary process, performed by the systems of FIG. 1, for transferring state data according to certain embodiments of the present invention.

Detailed Description of the Invention

As used herein, the term "hypermedia" is meant to refer to any kind of media that may have the effect of a non-linear structure of associated elements represented as a network of information-containing nodes interconnected by relational links. Hypermedia is meant to include

"hypertext", and the two may at times be used synonymously in the broad sense, but where stated or otherwise clear in context, "hypertext" can refer particularly to text content, and "hypermedia" to extend that to content that includes other formats such as graphics, video, and sound. The terminology used herein is meant of be generally consistent with that used in World Wide Web Consortium (W3C) recommendations.

The associations of elements may be specified as "hyperlinks" or "links," such as described by the XLink (XML Linking Language), SMIL (Synchronized Multimedia Integration Language), HTML, XHTML, and similar W3C recommendations. Links define an association between a "starting resource," the source from which link traversal is begun, and an "ending resource," the destination, collectively referred to as "participating resources." A "resource" is used to refer to any addressable unit of information or service and may at times refer to a resource portion rather than a whole resource, and a "content resource" to refer to any resource suited to presentation to a user. In the context of hypermedia, "node" may be used synonymously with resource. "Navigation" is meant to refer to the process of following or "traversing" links. Unless specifically indicated as "link navigation" or otherwise clear in context, navigation also is meant to include the control of presentation within a resource, such as scrolling, panning, and zooming, using VCR-like controls to play a continuous media resource, and the like. Addresses for Internet resources are typically in the form of Universal Resource Locators (URLs) or Universal Resource Names (URNs) or other Universal Resource Identifiers (URIs), but may be based on any other suitable addressing mechanism. Hypermedia resources may contain content (also referred to as mediadata) and metadata (including hyperlinks), aspects of a resource may be declarative (such as markup) or procedural (such as embedded logic or program code elements) and may include embedded resources.

Links may have information about how to traverse a pair of resources, including direction and application behavior information, called an "arc," and such information may include link "elements" having "attributes" that take on "values." Behavior attributes include "show" to specify how to handle the current state of the presentation at the time the link is activated, "external" to specify whether the link is to be opened in the current application, or an external application, such as one suited to a special media type, "activate" to specify whether the link is triggered by some event, typically user interaction, or automatically traversed when its time span is active, and "target" to specify either the existing display environment in which the link should be opened (e.g., a SMIL region, an HTML frame or another named window), or trigger the creation of a new display environment with the given name. It should be noted that the term target is sometimes also used in the art to refer to an ending resource as the target of a link, as for a "target resource" or "target page."

Links may be contained in the starting or ending resource, "outbound" or "inbound" respectively, or may be independently stored as "third-party" arcs. Standard HTML links are typically outbound, but inbound and third-party link arcs may be useful, such as for adding links that are external to read-only or third-party content. By providing such external, third-party links, resources not originally intended to be used as hypermedia can be made into hypermedia. Third-party links may be collected in "linkbases." Linkbases may be directly associated with their starting resources by a resource that leads to both the starting resource and the linkbase, referred to herein as "coupled" linkbases, such as a set of image map links in a Web page that has an embedded image link, or may be "decoupled" and obtained by other means.

Where so indicated or clear in context, the term hypermedia may also be used to include "hypermedia-like" resources and systems that do not use coded links as such, but which support

functionally similar non-linear resource relationships using other more or less similar mechanisms, such as special coding and logic that implements structures such as menu structures that have a defined graph structure, transaction request forms that have an associated address or other process identifier for transaction submission, and selectable content elements having a defined relationship to other resources or actions. Use of VCR-like or audio recorder-like controls to add non-linearity to a linear medium (e.g., fast forward/reverse, and skip ahead), also referred to as "trick-play" functionality, is also considered as hypermedia-like.

According to embodiments of the invention, links may refer to specific portions of a node or resource, such a by an "anchor" that associates the link to a position in text (such as in a HTML "A element"), or an "area" or "region" that associates the link to a spatial portion of an object's visual display, or to non-spatial portions, such as temporal subparts that may be defined by "begin" and "end" attributes, also referred to as "time positions" or together as a "time scope" or "time-span." Similar facilities are provided by XPointer, which supports addressing into the internal structures of XML documents, and provides an "origin" function to enable addressing relative to third-party and inbound links. Unless otherwise indicated or clear in context, "anchor" may used herein to be synonymous with similar forms, such as origin and "area."

Hypermedia structures may also be understood in graph-theoretic terms, and modeled as a directed graph, consisting of a set of abstract "nodes," the resources, joined by directional "edges," the hyperlinks. In this usage, a linkbase defines a directed graph.

As used herein, and consistent with the Dexter Hypertext Reference Model, a "hypermedia system" allows users to create, manipulate, and/or examine hypermedia, and consists of a "run-time layer" that provides tools for accessing, viewing, navigating, and manipulating hypermedia, a "storage layer" that models the basic node/link or resource/link

network structure of the hypermedia, and a "within component layer" that addresses the structure of components or resources of various given types. The storage layer, as used herein, includes media that may be streamed directly from a media capture device, such as a camera, microphone, or other sensor, and may not actually be stored. "Streaming" refers to this process of transmitting a resource representation, whether or not the resource is stored or not, and the representation may be in a format suited to storage, or one specifically suited to streaming. A data stream may itself contain multiple data streams, including both continuous media streams and other kinds of data or resources, including discrete resources, metadata, and the like. Depending on the particular embodiment, streams may contain channels, or channels may contain streams. Linkbases associated with streamed media may also take the form of continuous metadata streams, whether embedded with the mediadata stream or as an independent stream.

As used herein, a "browser" or "media browser" is meant to include any kind of presentation system capable of presenting media, and is used synonymously with "user agent" as a process within a device that renders the presentation data for a resource into physical effects that can be perceived and interacted with by the user. A "hypermedia browser" includes browsers that support hypermedia, including standard Web browsers, SMIL players, interactive television presentation systems (including self-contained advanced TVs and TVs with set-top boxes), and the like, and specialized applications capable of presenting hypermedia, including word processors, multimedia and video editors, virtual reality presentation systems, game players, and the like. "Player" or "viewer" may be used as synonymous with browser, and use of any media type descriptor as an adjective with "browser" refers to a browser capable of that media type. Thus any conventional TV set is included as a "browser" or a "TV browser," and

music players and radios are also included as browsers unless otherwise indicated or clear in context. Cases where hypermedia functions are not used are referred to as "linear" or "simple" presentation, viewing, or listening. "Media player" is used to refer to all such players collectively. Similarly, "browsing" is used to refer to any kind of viewing or playing experience, inclusive of hypermedia browsing and simple or linear viewing (such as watching TV), unless otherwise indicated or clear in context.

Web browsers are commonly limited to read-only use, except perhaps in use of forms, but other hypermedia systems are not so limited, and as used herein, unless otherwise stated, such as by the term "pure browser" or clear in context, "browser" is meant to include systems capable of resource creation and editing as well, including sound and video editing. Key functions of a browser include, but need not be limited to, providing access to resources, presentation of resources to the user and navigation of hyperlinks under user control or as directed by the hypermedia resources and links.

"Presentation" is meant to include any means of making a resource sensible to a human user, including visual display and audio, as well as any other sensible presentation such as used in current and future virtual or augmented reality systems affecting the sight, sound, touch, haptic, smell, taste, motion sensing, heat sensing, neural or other physiological interface, and the like. In addition to such "output," presentation also includes recognizing and responding appropriately to user "input" and/or "signals" of any kind that may be provided for, including keyboard, character recognition, touchpad, pointing device, haptic, microphone/speech, and camera, as well as more exotic inputs such as gesture, body movement, brain wave/electroencephalogram, neural or other physiological interface, and the like.

"Media format" or, synonymously, "resource format," as used herein refers to the format of a resource as retained, or potentially retained, as when streamed, in the storage layer and accessed by the browser, including access from local storage, via communications from a remote storage location or server or as streamed from storage or a live capture source. "Presentation format" refers to the format as rendered or otherwise processed by a browser or equivalent viewer or player or presentation system for actual presentation to a human user in sensible form.

"Hypermedia system" as used herein refers broadly to all system elements comprising such a system, including the hardware, software, communications, and storage, including portions at a user location, portions at server/peer locations providing content and processing services, potentially including the entire Internet or any similar network to the extent that those elements are usable with a hypermedia presentation system and the resources that may be accessible to it. "User system" refers to the portions local to or controlled by an individual user or a group of users of a shared presentation system. "Server" or "server system" refers to any system, whether hardware or software, providing auxiliary services that may be supportive of a user system. "Remote servers" include content servers or repositories, application servers that may perform information processing, searching, e-commerce, or other transaction or support services remote from the user, including TV and video servers, audio servers, other storage servers, including storage area networks (SANs), network addressable storage (NAS), game servers, virtual reality servers, cable and satellite TV and ITV head-end systems, network servers such as proxies and caching servers, and the like. "Head-end server" is meant to be inclusive of other remote servers that may be reached via the head-end, regardless of actual location or function. "Local servers" include analogous services that may be local to the user, including media servers, gateways, controllers, PCs, hubs, storage servers, storage area networks, DVRs

(also referred to a PVRs). Peer systems may also provide services in "peer-to-peer" (P2P) systems, and unless otherwise indicated or clear in context, the term server is meant to include peers acting in service provider roles.

"User" as used herein refers to any human end-user of a system, and may include users of a shared system. Users may be private consumers or workers in an organization or enterprise. User and "viewer" may be used synonymously. Depending on context, "subscriber" may refer to a user of a subscription service or more loosely to any user. "User interaction session" or "user session" as used herein refers to a series of interactions with a hypermedia system by a user, especially a series having a degree of continuity and relationship in time and with regard to an activity workflow or series of workflows, including concurrent workflows that may be related by a multitasking user.

According to embodiments of the invention, a user session may be composed of one or more "browser sessions," and well as other "application sessions" with other applications. The relationship of such sessions with each other within a user session may vary with different embodiments and with the settings and circumstances. For example, with enhancements to a TV-centric browsing experience, it might normally be appropriate that the base TV program and the related enhancement session be considered as "linked sessions" or sub-sessions that are distinct from one another, so that a browser session transfer is understood to transfer the enhancement session, but not the base TV session. The terms "transfer" and "migrate" are used synonymously to refer to the movement of the locus of work of a session, such as from one system or device set to another. The term "clone" is used to refer to a transfer that duplicates the current resource presentation of a session at a second device set. A migration that deactivates the session at the original device set is referred to as a "complete migration" or "terminal migration."

A user session may be local to the user system or may involve one or more "communications sessions" with remote server or peer systems, where such communications sessions may be defined in accord with a communications protocol. A user session may be composed of multiple "client/server sessions" (or "peer sessions" or "client/server/peer sessions", or collectively "remote sessions"), including concurrent such sessions. A "server session" refers to a series of activities performed by a server in support of a series of client/server service requests (and similarly for a "peer session" and "remote session"). Except where indicated otherwise or clear from context, references to peer-to-peer and client/server are meant to be inclusive of one another. Some protocols, such as HTTP for example, may be sessionless (based on request-response sets only), so that a remote HTTP communications session may strictly speaking be composed of multiple separate communications interchanges at the protocol level that are related by the server into a single server session, and this can be thought of as constituting a single virtual communications session. Unless otherwise stated or clear from context, communications session is meant to include such virtual sessions.

"Shared sessions" or "multi-user sessions" are applicable to multi-user systems where users cooperate or collaborate in controlling an interactive session, are recognized as individuals, and retain their individual identity and state.

"State" refers to the representation of the current state of a system relating to one or more tasks or sessions, usually in discrete values of some set of "state variables" that can be stored as a "state record" sufficient to define the state fully enough to allow the current activity to be deactivated and then reactivated, such as in a context switch or shutdown, using the state record to reset it so that it then behaves as if never interrupted. "Session state" refers to the state of a user session, for a browser session typically including, depending on the granularity desired, a

selection of such state variables as the user identity and related authentication information (including for example password and certificate information), the identification of active hypermedia resources and details of how they are currently rendered (such as window sizes and locations, and scrolling state), link arc data for any link currently being traversed, the execution state of embedded logic components such as Java applets (including the state of a Java Virtual Machine, JVM), ActiveX controls, Javascript (or ECMAScript, or Jscript, or other scripts), or FLASH, or other plug-ins, or helper applications, or the like, navigation path history (the ordered list of resources back and forward from the current resource, corresponding those activated by the back and forward browsing controls, as well as, optionally, next and previous with regard to tree branches), selected interaction history, variable user preferences, status of communications and server/peer sessions (including addresses, ports, identities, and authentication information), and other current context regarding other internal and external resources, including such information as may be stored in cookies. Any or all of such information may be stored in a "state record." State records may include details of user interactions not yet saved in the storage layer, including edits and forms field inputs not yet submitted. State may also include data on link arcs, including trigger data, and on resources, if such data must be transferred to establish state in a coordination embodiment in which such information cannot be obtained directly by the coordinated system. Sessions, software processes, and the like that are characterized by state variables are referred to as "stateful" and those that are not, as "stateless."

"Software process state" refers to the program environment state of a software process as it runs on a system. A process typically runs with the support of an operating system, and its state typically includes the current values of the instruction counter, registers, dynamic memory, input/output activities, and open or assigned operating system, network, and hardware resources,

as well as active sessions with external systems, and is used synonymously with "task," as an operating system concept that refers to the combination of a program being executed and bookkeeping information used by the operating system. Note however that "task" is more commonly used herein to refer to tasks at the user and/or session level. A software process is meant to include any of application software, middleware, and system software, and the case of a pure hardware, firmware, or dedicated implementation is also meant to be included in this usage.

A "process instance" refers to a single process with its associated state information. It may be possible to run multiple browser process instances on a single computer, sharing some system resources, such as caches, persistent storage, network access, and the like, in common, and thus having some state elements in common. Depending on implementation, a browser instance may allow for multiple presentation windows to be open, each presenting a different resource (and, as for example in MSIE, each supported by a separate process thread within one browser process). In such cases, depending on context, browser state may refer to the entire set of state information for all active browsers or the information for one browser instance (also referred to as one browser), for all its active windows. The term "current state" may be used to denote more limited state information on the single window, or single browser instance that is currently in focus for user interaction.

"Context" may be used as generally synonymous with state in referring to the information needed to allow a session to be interrupted, moved, copied, restarted, or otherwise shifted without apparent loss of context beyond the intended change. Context may also be used to refer to broader aspects of state that go beyond and are external to the state of the application, hardware, software, and network, to include the user, both in regard to his session, and potentially to the broader situation and environment of the session, including aspects that may be

sensed or inferred. This broad usage of context is defined (by Dey in "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications") as: "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects." "Context-aware applications" refer to those that exploit this broader class of external knowledge of "where," "what," "when," and "who."

"Client/server state" refers to the aspects of state relevant to a client/server session between a client system and a local or remote server system that provides it with resources or other services. "Server state" refers to those portions of client/server state maintained at a server, and "client state" to those maintained at a client (and similarly for "peer-to-peer" sessions). "Transaction" is meant to broadly include any discrete activity, but with emphasis on activities such as database inquiry, search, and update, which may or may not relate to business transactions, especially those that involve client/server (or peer) interaction and that may involve multiple processing, database update, and intermediate interaction steps.

"Granularity" of state refers to the level of detail captured as state and thus determines the number and kind of discrete points at which is can be saved and restored without loss of context or need for the user to re-establish lost context details. Examples of varied granularity include the relatively coarse grain of browsing link traversals, the intermediate level of user interactions for editing, data entry, and manipulation of controls and the like, and the very fine grain of internal software process state. The later is of lesser concern for much of the present

work, so that the granularity of user input, which is "relatively fine" in comparison to link traversals, may also be referred to herein as "fine grained."

"Interactive Television" (ITV) as used herein is meant to refer to any combination of video with displayable supplementary information and/or control elements that invite or aid in user interaction, including Enhanced TV (ETV) (or Extended TV), Synchronized TV (SyncTV), and similar services, and all forms of hypermedia containing a significant video component. This may broadly include the full range from "TV-centric" media in which the video program is expected to be the core experience in which interactive enhancements and features serve as complements, to "PC-centric" or "Web-centric" media in which computer-based media such as Web pages are the core experience and video serves as an enhancement or offshoot to that, but as may be stated or clear in context, ITV may be used to suggest TV-centric media. It is also meant to include specialized or more limited forms of interactivity with TV, including video on demand (VOD), near video on demand (NVOD), subscription video on demand (SVOD), pay-per-view (PPV), Enhanced (or Interactive) Program Guides (EPG/IPG), Digital Video Recorders (DVRs, also known as Personal Video Recorders, PVRs), Multi-camera angle or Individualized TV. Included are closed services such as "walled gardens" or "virtual channels" or ITV portals, and open services such as those based on Internet resources. More advanced forms of ITV include "viewer participation" capabilities, in which view interactions may result in changes to the program seen by other viewers, such as in polls or voting to select winners in contests, or even to alter the plot of a story ("interactive storytelling".) ITV includes systems using TV industry standards, such as ATVEF (Advanced Television Enhancement Forum) and the related DASE (Digital TV Applications Software Environment) and DDE (Declarative Data Essence), OCAP (Open Cable Application Platform), JavaTV, DVB-MHP (Digital Video Broadcast-Multimedia

Home Platform), DAVIC (Digital Audio Visual Council), ATM Forum, Interactive Services Architecture, or similar standard or proprietary systems (including for example ACTV/HyperTV, WORLDGATE, WINK, WebTV, and VEON, and the like), as well as Internet and Web standards, such as for example SMIL (Synchronized Multimedia Integration Language), MHEG (Multimedia and Hypermedia information coding Expert Group) and HyTime (Hypermedia/Time-based Structuring Language, ISO/IEC standard), and the like.

As used herein in reference to content resources, unless indicated otherwise or clear in context, "television" may be used as broadly inclusive of any video content or resource, including all forms of TV distribution, as well as movies, however distributed, live or recorded video, animations, 3DVR, or any other continuous visual media or audio/visual combinations.

Reference to "identity" of a "TV program" or for a radio program or other hypermedia resource external to the Web or an equivalently structured storage layer is meant to refer to resource identification information for any such resource, and identity of a "current" program may be limited to the channel (or equivalent) or may use a globally unique channel identifier, but may also include time-position information, such as a fixed time position from the start of a given segment, or a current position in real-time play, which may be specified in terms of a fixed position and a real time at which play begins from that position. "Identity," "program identifier" and "resource identifier" are used broadly to include any identifying information, including specific names or addresses or other unique program resource identifiers, including titles, naming codes, URIs, URNs, URLs and the like, MPEG-21 Digital Item Identifiers, TV Anytime Content Reference Identifiers (CRIDs), ISO/SMPTE/ATSC International Standard Audiovisual Number (ISAN) and Versioned-ISAN (V-ISAN), and relative identifiers, including time and channel identifiers. Such systems may distinguish between identifications used to logically reference a

resource, and locators used to actually retrieve the resource, possibly involving a resolution process and/or service to convert such identifiers to locators, and possibly supporting multiple alternative locations. The term "program" is meant to be used as broadly inclusive of any complete identifiable video (or audio or other media) segment or grouping of segments, including conventional broadcast or cable/satellite TV programs that may be identified by name or by channel and start time or other identifiers, as well as such alternatives as VOD or streamed programs from TV distribution industry or Internet sources, stored programs on cassette, CD, DVD, or other storage media or systems, and ad hoc programs such as might be obtained from a camera (or microphone) or computer-based image (or sound) generation source (such as 3DVR). Program is also meant to refer to advertisements, as just another class of program segment. The distinction between a program as a single resource and an interactive hypermedia experience as composed of multiple resources viewed in flexible, linked combinations may dependent on the context for cases where a program may involve some customization and/or personalization and variability in such aspects as multiple camera angles, sound tracks, short or long forms, and the like, and similarly as to whether advertisements are included or excluded as part of their surrounding programming.

As used herein, "user interface" (UI) refers to all aspects of facilitating man-machine interaction, including the hardware and software input/output (I/O) devices, and the control paradigms, models, and metaphors that exist in the user's mental model of the interaction, the real physical world, and the virtual world presented to the user as a shared conceptual medium that links the real, the mental, and the internal model of this world represented in the machine. "Graphical user interfaces" (GUIs) are widely used to facilitate user understanding and to implement virtual controls ("widgets") that may metaphorically represent physical controls (such

as a virtual button image on a screen). Less capable devices may be limited to simpler UIs based on menus and simple buttons. Multimachine user interfaces (MMUIs) refer to UIs that are capable of presentation on multiple machines having input/output devices and processors that are physically independent. This corresponds to the idea of systems, originally used with regard to data processing servers, that could be used independently, but in which software and network connections are used to give the effect to the user of a "single system image". Unless otherwise indicated or clear in context, MMUI is also used as a superset that is inclusive of the simpler cases of UIs that support multiple input/output devices driven by a single processor, including simple cases of multiple monitors, and of standard single machine UIs, and "full MMUI" or "true MMUI" or "independent MMUI" may be used to refer specifically to aspects or implementations that involve independent systems, and "multidevice user interface" (MDUI) may be used to more properly describe the broader, more inclusive use of MMUI. "Single machine user interface" (SMUI) may be used to refer to the case where no provision is made for a MMUI. "Machine" and "system" are used synonymously. Further clarification the usage of the term "independent" is provided in the discussion below

As used herein, "presentation device set" or "device set" refers to the input/output devices managed by a system as a related set for combined use as an access mechanism suite to support a user interaction session at a locus of work. Typically, independent systems have independent device sets. "Locus of work" refers to the spatial proximity of devices in a device set as related to the user, which can be thought of the "working set" of devices for a task, and device set and locus of work are used as roughly synonymous. "Lean forward" device sets refer to devices designed for intensive interaction and use in close proximity to a user, for "close work," such as PC devices, including display monitors, keyboards, mice, touchscreens, and the

like. "Lean back" device sets refer to devices suited to use at a distance, or "across-the-a room," such as TVs or music systems, and directly associated input devices, such as remote controls. In this usage, the locus of work for a device relates more to the perceived locus of its effect than its actual location, so, for example the locus of work for a remote control or wireless keyboard used to interact with a distant TV is primarily across-the-room, with the TV (as a projection of action to the TV), but secondarily in the user's hand. A screen is typically the dominating device, and other members of its device set will ordinarily have the same primary locus of work. For music systems, this locus is more diffuse, and the device set includes the speakers, the control devices, and microphones, if used. Similarly, voice input, gestures, or the like may have an ambiguous association with device sets. Specific commands or scoping conventions may be used to selectively direct voice commands (and similar ambiguous inputs) to specific device sets, systems and application components. "Physical locus of work" refers to the actual device sets and form factors as just described, while "logical locus of work" is meant to refer to the context of a session, and especially the presentation features, such as navigation position and essential aspects of resource presentation that a user could reasonably expect to be invariant after a well-effected transfer of physical locus.

As used herein, "coordinated" systems or device sets are those that are operated as an ensemble, in a coupled manner using the methods of the kind described herein or other similar methods. Such coordination or coupling may range from tight to loose, as described herein, and tight coordination may include synchronization. Coordinated devices sets may or may not be controlled by independent systems. Device sets that are recognized as being available for coordinated use at any given time are referred to collectively as a "device set group" or simply a "device group."

Except where indicated otherwise or clear in context, terminology used herein is meant to be generally consistent with that used (with respect to the Web) in Device Independence Principles (W3C Working Draft 18 September 2001), and specifically including the terms listed in its glossary, and with that used in the W3C Multimodal Interaction Working Group Charter (2/1/02), and in Multimodal Requirements for Voice Markup Languages (W3C Working Draft 10 July 2000). A notable area of variation from W3C convention is that, except where indicated otherwise or clear in context, "mode" is used herein as inclusive of differences with respect to work style modes and device set usage modes, such as lean-back versus lean-forward, which primarily relate to form-factor, as well as of differences with respect to sensory mode, especially speech, such as text (with image and pointing) versus voice (as supported by voice recognition or text-to-speech). The former are referred to herein as "homologous modes" and the latter as "heterologous modes." W3C usage of "multimodal" is specific to multiple heterologous modes, where one mode is a speech mode, and one is non-speech. Also in variation from W3C, "coordinated" as used herein includes cases of sequential coordination of device sets over the course of a user session, such as by session transfer, which W3C usage could refer to as uncoordinated using their definition of coordinated as being interpreted together (with regard to heterologous multimodal inputs or outputs). That narrower usage is referred to herein as "coordinated interpretation" or "synchronized coordination." "Personalization" is meant to include any process for user control of how resources are presented or used, both before the fact and at the time, including controls at a server or proxy at an application or adaptation level or in the browser or other associated user agent components, including selection of profiles that may be created by others. "Customization" is meant to include personalization as well as similar processes and controls that may be specified by an author/producer.

"Form factor" as used herein is used to broadly characterize the ergonomic or human factors aspects of size, shape, and configuration of a system and its input/output device set, primarily with regard to hardware characteristics unless otherwise indicated. "Adaptation" of a presentation refers to changes associated with different form factors of the device sets used. "Basic adaptation" refers to changes inherent in the form factor, including changes in display resolution and color depth, as well as the related issues of input devices relating to keyboards and pointing devices. "Rich adaptation" refers to substantive changes in the nature of the user interface such as use of menus, icons, text, and controls suited to high or low resolution display and varying abilities to enter text and control complex widgets such as drop down lists (but still with regard to homologous modes). "Heterologous adaptation" refers to the still richer adaptation to differing heterologous modes, such as speech.

"Author" is used to refer to any or all of the original author or creator of a resource, and editor or producer or programmer, or system operator, or other participant in the resource creation and distribution process (including advertisers, advertising agencies, and sponsors, in the case of resources which involve such parties), and thus inclusive of both content creators and content providers. As will be clear from context, "programmer" may be used to relate to TV content programs or to software code programs. Compound forms, such as "author/producer" or similar combinations, are meant to be synonymous with this inclusive use of author, and not to exclude unnamed roles unless otherwise indicated or clear from context. "Operator" or "system operator" or "service operator" is meant to refer broadly to operators of a TV distribution system, including Multiple System Operators (MSOs), TV networks, local broadcast stations, cable and satellite TV operators, as well as operators of Internet-based or other new channels of distribution (such as streaming media services), and of physical media distribution channels

(such as CD and DVD). Author is also meant to be inclusive of both "human authors," including any human editing processes, and "automated authors," including dynamic content management/delivery systems, software agents, association, filtering, and annotation systems, and the like.

Notwithstanding the distinctions made herein among TV, PC and other classes of user systems (such as listed below) that relate to such issues as the type of media they are oriented to present and to separations of reception, control, and storage functions, it should be understood that such distinctions are not inherent or essential to the methods described, and will gradually dissolve as these products continue to converge. These distinctions are used to address current and near-term product configurations, and not to imply restrictions in the applicability of the methods described. Thus for example, TV and PC, TV and STB, TV and DVR, and similar currently disparate configurations should be understood to be synonymous with regard to future fully converged products.

"Television system" (TV) or simply "television" as used herein refers to a system for presenting video, whether from a transmitted or stored resource, and unless stated or otherwise clear from context includes reception and control components such as typically contained in a TV "receiver," as well as advanced control, reception, and storage functions which may be separately contained in a "set-top box" (STB) (but not necessarily including advanced media gateway and server functions that may be packaged together with a set-top box). Also included are associated input devices, such as remote controls, and storage devices such as VCRs (Video Cassette Recorders) and DVRs. "TV-like" or "TV-type" are used to refer broadly to all systems having a predominant function of playing video. Unless otherwise indicated or clear from context, set-top box or STB is used both to refer to a separate set-top box unit, and to include the

equivalent functions (control, signal management and conversion, intelligence, and the like) that may be integrated into an advanced TV system or receiver, as well as possible future configurations that may combine STB functions with a gateway or other system or that may distribute such functions into multiple units to control multiple TV receivers or monitors or other display systems. In this broad sense, STB and TV may be used interchangeably.

"Computer system" or simply "computer" when used herein in the context of a user system, refers to any kind of intelligent system used predominantly as a general purpose intelligent device capable of running "application programs" for various purposes. A variety of conventional distinctions may be used to categorize computers as to functional capability and form factor, such as, for example, those listed in the next paragraph, but such categorization should be understood to be fuzzy, and likely to evolve over time as capabilities change, improve, and converge and usage patterns co-evolve. "Computer-like" or "computer-type" are used to emphasize inclusion of all such systems and exclusion of systems where computer function is absent or subordinate to television functions. User systems that do not predominantly function as general-purpose computers may nonetheless contain "embedded computers" to provide supporting intelligence, such as for example, in media players or other entertainment devices.

Personal computer" (PC) may be used to refer broadly to any computer for personal or individual use, but as will be clear in context, usually suggests a desktop or laptop/notebook (or sub-notebook) form factor that provides for a high-function, high-resolution user interface. "Personal Digital Assistant" (PDA) refers to a wide range of handheld and portable devices that provide PC-like capabilities in a reduced size and weight form factor, typically with small screens and no keyboard. "Tablet" may refer to a complete system that provides an intermediate form factor, with a screen, and a touchpad or stylus interface and possibly including a compact

keyboard, but can also refer to a similar device that serves as terminal to a base system. Additional computer-like systems are Internet appliances, wireless phones and pagers, which are gradually converging with PDAs.

It should be understood that advanced TV/entertainment device remote controls may include display screens and stylus or touchscreen entry that is comparable to a PDA in form factor, and that PDAs typically have infrared communications and may be used with software that can enable them to serve as TV/entertainment remote controls. "Dedicated" is used to refer to devices are designed to work with a specific class of associated devices, especially those with a specific architecture, and which may generally be expected to be "provisioned" together, and "non-dedicated" or "open" to those designed for flexible use and interfacing to a wide variety of system types and architectures. Such dedicated devices may commonly also be "limited function" devices, lacking the "general-purpose," open programmability typical of a PC or PDA, a capability that allows for an open-ended range of applications. As a result, dedicated devices may be limited in utility and unable to achieve the economies of scale and breadth of function of more flexible platforms. "Universal" may be used to refer to the very partial step of a device such as a "universal remote control" that is designed for use with a limited class of associated devices (in this case TVs and other entertainment devices) from any of multiple vendors, but which lacks broader function (in this case use as an independent PDA).

As used herein, the term "continuous media" is meant of refer to any representation of "content" elements that have an intrinsic duration, that continue (or extend) and may change over time, including one or more of "audio data," "video data," animation, virtual reality data, hybrid natural and synthetic video data, including both "stored formats" and "streams" or streaming transmission formats, and further including "continuous hypermedia" which contain both simple

continuous media and hyperlinks. Continuous media may contain descriptive metadata, time codes (such as in Society of Motion Picture and Television Engineers, SMPTE, or European Broadcasting Union, EBU, coding), and other metadata. Resources that are not continuous, and have no temporal dimensionality are referred to as "discrete." Continuous media is also inclusive of "time-based documents" as used in the HyTime standard to refer to documents with scheduled presentation. "Time code" is meant to include specific time code values embedded in the video, such as SMPTE/EBU, or other signal data that can map to exact time positions, as well as external measures of time position that may or may not be exact, including for example such timing systems as are used in SMIL and MIDI.

"Video data" refers to all forms of moving -images, with or without accompanying sound, including analog and digitally coded video, television, Internet television or IPTV or IP video, film, animation, virtual reality data, hybrid natural and synthetic video data, and the like. Video image data is most commonly represented as a series of still images, whether in analog or digitally coded forms, including ATSC (American Television Systems Committee), NTSC (National Television Systems Committee), PAL (Phase Alternate Line)/SECAM (Sequential Couleur avec Memoire), DTV (Digital TV), HDTV (High Definition TV), EDTV (Enhanced Definition TV), SDTV (Standard Definition TV), MPEG (MPEG-1, MPEG-2, and MPEG-4, and supplemented by MPEG-7 and MPEG-21, and other standards), DVB (Digital Video Broadcasting), International Telecommunications Union H.26x and H.32x, RTP (Real-Time Transport Protocol), RTSP (Real Time Streaming Protocol), SMIL (Synchronized Multimedia Integration Language), ISMA (Internet Streaming Media Alliance), QUICKTIME, WINDOWS MEDIA, and REALMEDIA, and the like, but may also be coded as object data, including formats provided for in MPEG-4.

"Audio data" refers to all stored forms of sound, whether part of a video form or not, including analog and digitally coded sound or music or other audio information in formats such as PCM (Pulse Code Modulation), CD-AUDIO, MP3, REALAUDIO, MIDI (Musical Instrument Digital Interface), and the like. Audio data is most commonly represented as amplitude data over time, whether in analog or digitally coded form, although object data representations can also be, such as using MIDI.

Animation or virtual reality data is commonly represented in various image-like forms, raster or vector graphic forms, or as object-based structures, such as scene graphs, including SHOCKWAVE FLASH (including SWF and Open SWF), SVG (Scalable Vector Graphics), VRML (Virtual Reality Modeling Language), RM3D (Rich Media 3D), X3D (eXstensible 3D), and MPEG-4/BIFS (Binary Format for Scenes), Computer Aided Design (CAD) or wireframe animation, and the like. Unless otherwise indicated or clear from context, "virtual reality" is meant to be inclusive of augmented reality.

Another media content type is still images, including photographs, drawings, cartoons, diagrams and facsimiles, which may be coded in such formats as JPEG (Joint Photographic Experts Group)/JFIF(JPEG File Interchange Format), GIF (Graphic Interchange Format), TIFF (Tagged Image File Format), PTP (Picture Transfer Protocol), including object formats such as CAD and the other object formats listed above, and the like.

A further common media content type is text, which may be coded in such formats as ASCII (American Standard Code for Information Interchange), HTML (Hypertext Markup Language), DHTML (Dynamic HTML), XHTML (eXtensible HTML), PDF (Portable Document Format), SGML (Structured Generalized Markup Language), Postscript, word processing

formats, and the like. Other media content includes active formats, such as spreadsheets, for example.

"Media content" (or "media") is used herein to refer generally to any content, or information that is understandable to humans. "Content" refers to any form of transmitted or stored information. "Objects," when used in the context of stored content objects refers to any content item or element or grouping of items or elements, including objects within a file, and objects stored as files or sets of files. When used in the context of object-based media formats, the term is meant herein to be used in accordance with the definitions applicable to such formats.

"Storage" as used herein is meant to refer to the process of storing information or content for future use, or to any memory, "storage device" or "storage system." "Storage system" refers to any device or any combination of one or more devices with software that supports the use of storage, including SANs and NAS. "Storage device" refers to the element or elements of a storage system that include actual fixed or removable "storage media" capable of retaining content in an electromagnetic or other machine-readable form using any technology, including electronic, magnetic, optical, time-delay, molecular, atomic, quantum, transmission-delay and the like, including all future storage technologies.

"Transmission" as used herein is meant to refer to any form of "communication" or "transport," including connections to directly attached devices, local area networks (LANs) including home and office networks, and wide area networks (WANs). Transmission may be over any suitable medium, including the Internet and World Wide Web, cable and wireline networks, including DSL (Digital Subscriber Loop) telephonic, Hybrid Fiber/Coax (HFC), powerline or others, ATM (Asynchronous Transfer Mode) networks, fiber-optic networks including use of SONET (Synchronous Optical Network) and DWDM (Dense Wavelength

Division Multiplexing), satellite and terrestrial fixed and mobile wireless networks, including broadcast, direct-to-home (DTH) satellite, cellular, 3G (3rd Generation), future 4G or NextGeneration, UMTS (Universal Mobile Telecommunications System), LMDS (Local Multipoint Distribution Services), MMDS (Multipoint Microwave Distribution System), and wireless LANs (WLANs) such as IEEE 802 series (802.11a, 802.11b, 802.11g, 802.16) wireless Ethernet or Wi-Fi networks, ETSI HiperLAN, and other wired or wireless LANs and HANs (Home Area Networks) and PANs (Personal Area Networks) or WPANs, including Bluetooth, HomeRF, infrared (including IrDA, Infrared Data Association), powerline, including HomePlug (HomePlug Powerline Alliance) and X10, phoneline, including HomePNA (Home Phoneline Networking Alliance), and variations based on Software Defined Radio (SDR) and spread spectrum methods, as well as ad-hoc networks. Unless otherwise indicated or clear from context, LAN, HAN, and PAN (and their wireless variants) are meant to be substantially equivalent and inclusive on one another. Transmission includes direct (point-to-point) wired paths, including special purpose local connections using proprietary or standard physical and signaling methods, including audio/visual (A/V) connections such as baseband video, channel 3/4 ATSC RF, RF bypass, S video, baseband audio, and SP/DIF digital audio, cable connections, twisted pair, Universal Serial Bus (USB), IEEE 1394 Fire-wire, and the like, as well as wireless equivalents such as wireless 1394 and infrared. Unless otherwise indicated or clear from context, transmission is meant to include physical transport of storage media. Transmission involves both a logical path, which is meant to refer to higher-level protocol and routing considerations, and a physical path, which relates to the lower level of the specific wired or wireless media signaling paths used. Transmission may be one-way, such as broadcast, or two-way. Two-way cable television networks may provide for a return channel that is in-band or out-

of-band, or may use telephone lines and modems to achieve similar return connectivity, thus supporting push or pull activity.

Transmission or network protocols may include IP (Internet Protocol, including IPv4 and IPv6), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP (Stream Control Transmission Protocol), RTP, RTCP (RTP Control Protocol), RSTP, IP Multicast, ASF (Advanced Streaming Format), HTTP (Hypertext Transfer Protocol) and the secure variant HTTPS, Internet Relay Chat (IRC), Short Message Service (SMS), Simple Mail Transfer Protocol (SMTP), Jabber, T.120, WAP (Wireless Applications Protocol), ATM, Ethernet, GSM (Global System for Mobile Communications) and similar wireless protocols, cable TV and Hybrid Fiber/Coax protocols, DOCSIS (Data Over Cable Service Interface Specification), DSM-CC (Digital Storage Media - Command and Control), DMIF (Delivery Multimedia Integration Framework), and many other current and future protocols, and may use baseband or broadband signaling. In multi-node networks, transmission may be directed to a network node address, examples of which are IP addresses, STB or cable drop or satellite receiver node addresses, and logical addresses, such as URLs and URIs/URNs.

"The Internet" is meant to include both the current embodiment of the Internet with its current suite of protocols, services, nodes, and facilities, and future extensions (with extended protocols, services, nodes, and facilities) as an open internetwork that links and subsumes all networks that are not intentionally isolated from internetworking, including a multinetwork that uses an adaptation layer to bridge networks having diverse protocols. Unless otherwise indicated or clear from context, the Internet is meant to be inclusive of other networks or sub-networks using similar technologies or providing similar services, including intranets or extranets or ad-hoc network assemblages.

"Metadata" refers to data about data, including descriptors of data content and of data format and "program information." Metadata formats include XML (eXtensible Markup Language), RDF (Resource Description Framework), SDP (Session Description Protocol), SAP (Session Announcement Protocol), MIME (Multipurpose Internet Mail Extensions), MPEG-7, MPEG-21 (including Digital Item Declaration, Digital Item Identification and Description, Content Handling and Usage, Intellectual Property Management and Protection, Terminals and Networks / Digital Item Adaptation, Content Representation, and Event Reporting), SMPTE/EBU time codes, QUICKSCAN addresses, ATSC-PSIP (ATSC-Program Service Integration Protocol), DVB-SI (Digital Video Broadcast-Service Information), and SMIL, as well as data contained in Electronic Program Guides (EPGs). Metadata also includes markup, such as that used to define the presentation and handling of content, including link arc data, and markup is a coding method that can be used to express metadata. Unless otherwise indicated or clear from context, reference to XML is also meant to include use of the expanding suite of tools for working with XML including XLink, XPointer (XML Pointer Language), XPath (XML Path Language), XSL (Extensible Stylesheet Language), XSLT (XSL Transformations), Namespaces, Document Object Model (DOM), XML Information Set, XML Fragments, Canonical XML, and XML Schemas and DTDs (Document Type Definitions), XML Query, and ongoing enhancements to these tools and standards, as documented by the W3C, as well as other tools related to that work.

"Multicast" as used herein is meant to refer to the transmission of data to a defined group of recipients. Internet multicast protocols, such as supported by the Internet Multicast Backbone (MBone) and IP Multicast, provide for this in the form of a stream or channel to which recipients may subscribe. "Broadcast" is meant to apply broadly to any form of distribution intended to go

simultaneously to many recipients (one-to-all, one-to-many), including conventional TV and radio terrestrial broadcast, cable and satellite distribution, and the like. Unless otherwise indicated or clear from context, broadcast is also meant to include other forms of simultaneous distribution, whether true broadcast (one-to-all) or equivalents, such as realtime Internet streaming, whether using multicasting (one-to-many), or simultaneous unicast via multiple direct individual (one-to-one) sessions.

It should be noted that "synchronization" is used in two different senses, which will be clear from context. One relates to synchronization of usage activities among device sets, as a high degree of coordination, such that events at one device set are fully replicated at another for some continuing period. The other usage relates to synchronization of resource presentation, where the originating resource of a link is time-bounded, so that a link is enabled at the start of that interval and disabled at the end, such as is often desired in ITV, where presentation of video resources and associated enhancements are intended to be synchronized, so that an enhancement appears at the same time as a corresponding video segment. Further terminology related to timing of enhancements is suggested by Behrens, Prototypes, Field Tryouts Proceed For Enhanced TV in Current, July 17, 2000, and usage herein is meant to be generally consistent with that:

- "Synchronous enhancements that are transmitted for use at specific times in a program." (This may also be referred to as "program-synchronous.")
- "Always-on enhancements, such as navigation bars, that are constantly accessible at the click of a remote control or mouse."
- "Asynchronous or post-broadcast enhancements that are silently transmitted into the DTV receiver's memory and can be activated when the viewer chooses."

- "Interpolated (for lack of better word) enhancements that the viewer can choose to insert seamlessly into an ongoing program." (This causes the first program to stop, then resume after the enhancement, giving the effect of an insertion.)

Overview

With the above in mind, the present invention may be described, in various embodiments, as a system and method for navigating hypermedia using multiple coordinated input/output device sets. It provides a broadly flexible, powerful, selective, and simple user interface paradigm for browsing that allows the user (and/or an author) to control what resources are presented on which device sets (whether they are integrated or not), and provides an effective method for coordinating browsing activities to enable such a user interface to be employed across multiple independent systems.

One aspect is, in the spirit of human-centered design, to anticipate and be responsive to the user's desires (and the author's suggestions) as to what resources to present where, in order to make the best possible use of the hardware resources at a user's disposal. Homes, offices, and other personal environments of the future will have a rich array of computer-based input output devices of various kinds, some general purpose, and some more or less dedicated to specific uses. The desire is to minimize constraints on what system resources can be used for a given task, to enable the most powerful browsing experience possible. Browsing of hypermedia, such as in the case of ITV is a task in which the use of multiple devices might be valuable because it may be expected to be a dominant activity, if supported effectively, and because of the disparity of UI issues between watching extended video segments and doing intensive interactions (such as with Web media) that may be more or less closely coupled with such video segments.

Prior work has generally not recognized that it is inherent in rich ITV and similar forms of video-centric hypermedia browsing to be best served as "two-box," multitasking experiences, at least much of the time, and the problem is not to squeeze it into one box (and fight over which box's functionality and form factor is better), but to enable effective coordination of both boxes. While the TV vendors and the PC vendors might fervently wish to offer a single system that meets the needs of ITV users, that is not an effective solution. If one assumes that an ideal level of coordination among device sets can be enabled and explores usage scenarios, it can then be seen that different modes of viewing are best served by different device set form factors. These modes are not fixed for the duration of a session or task, but can blend, overlap, and vary as the flow of a set of linked tasks changes. What begins as a TV-centric browsing (or pure viewing) experience may shift to casual use of a PC for light interaction (such as looking at menus and options or doing a quick lookup) to intensive PC-centric activity (and then back again). The user may shift focus from the TV to both, to primarily the PC for a time, then become involved in the TV again. Conversely, an user at a PC may shift to immersion in a TV program or movie, then return to intensive use of the PC. While some broad usage patterns tend to favor video on the lean-back TV device set and interactivity on the lean-forward PC-type device set, other issues may relate to incidental viewing of video from a PC centric phase of activity, and casual interactions with enhancements in a TV-centric experience, as well as a complex mix of secondary issues, such as quality-of-service factors, whether an alternative device set is at hand and ready for use, other activities, presence of other people, location/setting, mood, and the like. The point in a session at which a user may wish to shift device sets may depend not only on the immediate task, but the user's expectation of where that task is leading, so an intensive task soon to end may not warrant a shift from TV to PC, but a less intensive task leading to deeper

interaction may warrant an early shift. Varying form factors of different TV devices and of the range of PCs, PDAs, tablets, and Internet appliances may also affect what tasks a user wants to do on what device, with what UI. At the same time, to avoid burdening the user with the complications of too much flexibility and too many choices, it may be desirable that both the user and the content author be able to pre-set affinities, preferences, and recommendations, relating to task types, content types, and device availabilities, that could automatically place elements on the device set or device set group that is presumably best suited to the apparent context, while leaving the user with the ability to recognize that expected targeting (based on conventions and/or unobtrusive cues) and to accept it with no further action, or override it if desired.

Providing the desired flexibility can be viewed in terms of three interrelated issues, one of structuring an effective and flexible multimachine user interface (MMUI) for browsing by a user, one of providing methods (such as markup) for the resource creator/author/producer to aid in exploiting that MMUI, and one of implementing such an interface on a wide range of hardware and software, including systems for which such usage may not be a primary mission (including both new systems and legacy systems).

A general approach to a MMUI for browsing that provides both user control and authoring support may advantageously build on the concept of targets for presentation of linked resources already present in hypermedia formats such as HTML (and XLink). In HTML, the link target attribute can be used to specify which of multiple frames a linked resource is to be presented in, with options that include the current frame, another existing frame, or a new frame. Coded specifications within the link are typically set by authors/producers of content, and controls in the browser allow the user to override and alter these settings, such as (with

MICROSOFT Internet Explorer, MSIE) by using a shift-click combination to indicate that a link should be opened in a new window. Extending this to an MMUI can be done by expanding the coding of target attributes and by adding new browser control options, such as control-click, to target a window on an alternate device set. Additional control can be achieved by extending the richer drop-down control that is invoked in MSIE by right-clicking on a link. That drop-down list can be extended to list windows on alternate device sets. This provides a very flexible, general, and simple way to shift activity from one device set to another. Similar controls can be provided on simpler devices, such as for example, with a TV remote control, instead of select to activate a link to an enhancement overlay on the TV, a combination such as exit-select could be used to activate that link to an associated PC, or a new control button could be provided. As with current browsers, variations on such controls can also be defined to open the current resource at a second location (cloning).

To implement such an interface on multiple independent device sets, the ending system must be given information to inform it when a link is to be activated, to what resource, with what browser attributes, and with what context information. A basic method is to transfer from the starting system to the ending system a link activation message that that includes a state record and contains relevant link arc information. The state record contains essential information on the state of the browser and related activities on the starting system that can be used at the ending system to configure its browser and related context accordingly. A state exporter/importer/tracker component may be provided as an addition to a standard browser to provide these functions (with exporter/importer function being sufficient for simple applications).

In simple embodiments, export from the starting system and import at the ending system need be done only once per transfer of locus. In certain embodiments, full event synchronization can be maintained, when desired, by the state tracker to provide ongoing collaborative functionality, as well. This is useful in the case of multiple users, and also can be useful for a single user that desires the ability to use both device sets in a fully replicated mode. However an advantage of the proposed method over conventional collaboration and synchronization systems, is that such ongoing event synchronization is not needed for basic MMUI browsing by a single user, and the complications and overhead of continually logging, exporting and importing all events that may alter state can be avoided. Instead, state information need be assembled for transfer only when a transfer is actually invoked, and only at the necessary granularity. This simple, occasional, coarse-grained transfer is readily added to any browser of existing architecture, unlike more fine-grained full synchronization approaches, which require either excessive tracking activity, display replication approaches, or rearchitecting of browsing to use model-view-controller architectures, such as in event replication approaches.

Another key benefit of this method is that it is readily applied to heterogeneous systems with only simple addition of an exporter/importer and some new UI functions to each system's own native browser. This exploits the fact that the underlying resources being browsed can be common to all systems, and that at a high level, browsing state is relatively independent of system architecture. Thus the method is readily applied to both TV and PC-based systems, and could be added to existing or new systems by manufacturers, integrators, distributors, service providers, or by end users themselves. The proposed methods are well suited to standardization, which could facilitate the inherent capability of the methods described here to allow any suitably functional device sets and systems to be used together in the desired coordinated fashion,

regardless of its internal software and hardware architecture, vendor, or provisioning. Use of XML, RDF, and related standards is suggested to facilitate this. These features for ad hoc provisioning and use of devices acquired for other purposes removes a major hurdle to the introduction of MMUIs for ITV and other hypermedia browsing applications. Thus, for example, a household need not buy a lean forward device for ITV, but can simply use an existing PC, PDA, tablet, or the like.

Figures

Referring now to FIGS. 1 - 7, wherein similar components are referenced in like manner, various features for a method and apparatus for progressively deleting media objects from storage are disclosed.

Turning now to FIG. 1, therein is depicted a schematic of an exemplary home system environment 100, which with its key systems and device sets and related elements. A number of typically independent systems, are represented (having associated device sets not shown here in detail), including TV or ITV system 130, PC 140, and PDA and/or phone 150, and the like. The TV/ITV 130 is understood to commonly include a set-top box. These systems typically contain their own transient and persistent storage subsystems, not shown, and may share a common local storage system 160. These systems may connect to each other and the outside world via a home network or LAN (local area network) or hub 128, which may use wired and/or wireless technology. Auxiliary services may also be provided by a home gateway server, which may be combined with the LAN, STB, PC, or other device capable of acting as a server, and with other service components. External connections may be made directly from a single system, as shown for cable 122 connecting to the TV (STB), but may preferably be connected to a home network

to facilitate shared use by multiple systems, as shown for the connection to the Internet 124, and connection to wireless network 126 (which could also be an Internet connection, such as using Mobile IP). These external connections provide access to various servers and other sources for a variety of sources of content and connectivity 110, which may include broadcast, satellite, and cable TV, video on demand, IPTV, streaming media, Web content, wireless portals, transaction servers, and the like.

Referring now to FIG. 2, therein is depicted a more detailed exemplary schematic of typical TV and PC systems and associated device sets. FIG. 2a depicts the case of independent systems, showing home area network 128 connecting to both TV STB 210 and PC 220. TV STB 210 is used in conjunction with TV receiver or monitor 212 and remote control (RC) 214. The TV receiver 212 and RC 214 together constitute a device set, DS1. The controlling device TV STB 210 may for some purposes also be considered part of the device set DS1, and is considered together with the other elements of DS1 to constitute a system, S1. The PC 220 (more precisely the system unit) is used in conjunction with monitor 222 and keyboard 224, as well as other peripheral or input/output (I/O) devices such as a mouse, not shown. Those PC elements together constitute the device set DS2 (which may for some purposes also be considered to include the PC system unit) and system S2.

FIG 2b depicts an alternative case of an integrated system in which a single set of hardware takes the role of controller 260 providing functions of both a TV STB and a PC system unit, equivalent to devices 210 and 220, and thus represents a single alternative system S1. In this case TV receiver 262 and remote control 264 constitute an alternative embodiment of device set DS1, and the PC monitor 272 and keyboard 274 constitute an alternative embodiment of device set DS2. Again, the controller may or may not be meant to be included in references

those device sets, but it should be assumed to be excluded in references that distinguish the two device sets. In this special case, if both capabilities were fully developed such that one function was not clearly subordinate to the other, the overall system complex could be considered both TV-like and computer-like. The device sets could still be considered to be either TV-like or computer-like respectively, not both, and they could be considered not independent of one another with regard to processing, as described further below.

Actual embodiments can be expected to be determined by a complex mix of factors, only one of which is suitability to the browsing task. These factors include legacy equipment installed base constraints, industry tradition, vendor/intermediary/consumer market power, bundling, provisioning, policy, standards, regulation, and the like. Thus methods that are adaptable and broadly applicable to a wide range of configuration alternatives that may not be the ideal choices may be beneficial.

A key aspect of MMUIs is the concept of device sets, and the distinction of device sets is sharpest in the case of independent systems. A key attribute of an independent system is that it has its own processor(s), and is thus capable of running applications and driving UI device sets in a reasonably independent fashion, depending on the nature of the application and its dependence on external storage, network, and server resources. For example, an independent system is normally capable of running a "thin client," such as a browser, even if support for a "rich client" or "thick client" application may be limited. An equivalent hardware appliance with similar independent processing capabilities is also considered independent.

A closely related factor is whether a device in question is used as an I/O device between the user and a processor, or as an intelligent processor that is peer (or in a client/server role) to another processor. Subtleties arise when multiple intelligent devices are used together, such as

when an intelligent system acts in a role that makes it subordinate to another intelligent system, more or less as a simple I/O device, and when it acts in multiple varying roles. Thus independence can be a matter of degree, and can be more operative in some usage roles than others. A criterion that can be helpful in clarifying these cases is whether the system is sufficiently independent to be considered by the user as a separate computer, usable separately, or whether it produces the illusion of a single computer (e.g., that may have multiple devices sets attached).

In a hypothetical distributed configuration, for example, an intelligent tablet could be considered independent of a supporting PC if the tablet runs its own browser (such as under Windows CE) to obtain hypermedia resources in resource format (such as HTML) and render them into presentation format for display (such as display buffer image format), and could be considered not independent of the PC if it is driven as a replicated display by the PC, with rendering controlled by a browser at the PC and using a technology such as Windows Terminal Services (WTS) or the like, to transfer the resultant presentation to the tablet at the level of display buffer image format (or coded changes thereto). WTS and similar offerings from CITRIX use MICROSOFT Remote Desktop Protocol (RDP) or the closely related CITRIX Independent Computing Architecture (ICA), which are related to the ITU T.120 standard, to support a relatively dumb thin client that for the most part offers basic I/O terminal function only. Continuing this hypothetical, it should be noted that an independent structure alternative of this class could be the much simpler (and more efficient one) architecturally, in that a browser on the tablet can act as thin client directly to a Web server, using a base PC only as an intermediate network routing node, as opposed to an dependent structure in which a tablet operates as a WTS or CITRIX style thin client I/O terminal to an PC (WTS or CITRIX style) server, which in turn

mediates input and display I/O to support a browser on the PC (driven by that terminal) as a second level of thin client (to a Web server). It will also be understood that either of these PC-supported structures would still be independent of a separate TV system with which it might be used for coordinated browsing. In the same manner, such a tablet device could be supported or driven by a TV/STB system (instead of by a PC), with the same possibilities for dependent or independent structure alternatives, and potentially using the same protocols. If such thin client devices become popular, this might be an attractive way to add MMUI support to a TV/STB system, incorporating off-the-shelf terminal devices -- devices that a user might already possess or acquire for other uses as well.

A minor variation on this kind of distributed I/O theme is represented in systems like X Windows, which uses the X protocol to define a separation between an application and its display, with a windows manager that runs on a client and the application on a server (which X refers to as server and client, respectively, in reverse of the now common convention that is used herein) and in the I3ML (Internet Interface and Integration Markup Language) proposed by COKINETICS that applies somewhat similar concepts to distributing Windows UI controls. Here again, if the core browser functions of converting resources obtained in resource format to presentation display format is done at a server, this could be considered a dependent I/O device architecture with respect to the server, for purposes of this discussion.

These issues also relate to hypothetical configurations where an intelligent remote control might be used with a TV/STB system or other devices. A key factor is whether the device is used as an I/O device between the user and a processor, or as an intelligent processor peer to another processor. A dumb remote is considered part of the device set of the device it controls, so a multi-function remote used to control multiple systems may at times participate in multiple

devices sets. A smart remote, such as one based on a PDA might also operate independently, thus constituting a separate device set (and separate system) in that use. Thus a PDA acting as an independent but coordinated browser in conjunction with an ITV system could be considered independent and a separate device set in that use (being a peer processor), but to the extent that it also serves directly in emulation of a standard remote to the ITV system by sending standard remote control commands as activated by a user (as a simple I/O device), that use could be considered dependent in relation to the ITV system controller and thus part of the ITV device set.

These issues can also get complicated in some cases of a single system driving multiple devices. One instructive example is the use of a browser with a dual-display PC, and some hypothetical multi-monitor configurations are considered as examples of how the methods proposed herein might be embodied in selected cases. First, consider the basic structure of a single system controlled by the single keyboard and mouse, but having two directly attached monitors. This can be considered to be one "augmented" or "enriched" device set, as opposed to two separate device sets, since only a single input device set is used, and this is really just a case of adding more screen area of similar form factor. Further, at a software level, two monitors attached in such a way may actually be seen by the browser application software as a single monitor of double size, because standard Windows and APPLE Macintosh multi-monitor support provides for a virtual desktop the offers applications a mapping from a single extended virtual frame buffer seen by the application (browser) software to the two real frame buffers (corresponding to the primary and secondary monitors in Windows terminology) that drive the monitor devices. In such a configuration (whether using a virtual desktop or a separate, real display, an independent display in Windows terminology), coordination of browsing across the multiple monitors could be much simpler than for an independent system configuration with

independent device sets, because all browsing can be done using a single set of input controls to a single browser instance that simply controls the two display monitors in much the same way that it could ordinarily control multiple windows on the same display monitor. Such a single browser instance could have full, exclusive, direct access to, and control of, all browser state information, including all UI inputs and other I/O events, all caches and work areas, all storage, and the like, and thus could drive the two displays in the same way that it could drive multiple windows on a single display (if it could even see the two as separate displays). Thus in this case, basic support for simple targeting of alternate displays is a relatively simple variation from the existing function of targeting to alternate windows on the same display, and use of an exporter/importer transport function might not be required. The browser could simply control activity in the second display area by selecting windows on that monitor (or that portion of the virtual monitor).

A variation on such a case could occur if, as an added feature, the user were permitted to open a second browser process instance, and coordination across browser instances were desired. In such a case, the addition of export/import functions could be required, but this could be somewhat simplified in that much state information (such a page caches, history lists, and the like) may be in commonly accessible storage, and thus need not be included in the export/import process. This case is also simplified in that, being on the same machine, the two browser instances share common access to the hypermedia storage layer, and can communicate via intra-system means.

A further level of simplicity to be expected in such a hypothetical configuration is that, using standard multi-monitor support for a PC (or Mac), such displays must be functionally equivalent, driven as standard PC displays, with a possibility of only the minor differences in

size and resolution that is typical of PC monitors. This means that the rendering and presentation need not be adapted to deal with varying display characteristics (at least not beyond the basic levels of adaptation that might optionally be used at the server by highly tuned Web sites that sense a range of standard display resolutions using standard Web and browser support and adjust the pages served accordingly).

It will be understood that mirroring of displays, in which a display image is exactly duplicated on a second display, offers a related function that is widely available, and can be used to provide some basic capabilities in support of a multimachine-type UI, even though, as is clear from the teachings herein, it is generally desirable that the images in different device sets not be identical.

Specific components of the systems portrayed in FIGS. 1 and 2 tend to be somewhat divergent in current technology embodiments, and vary in accord with form-factor, but can be expected to increasingly converge toward similar or common technologies. These components include all of the usual elements of such systems, such as CPUs and other processors, clocks, various specialized logic processors, including, CODECs (Coder/Decoders), DSPs (Digital Signal Processors), ASICs (Application Specific Integrated Circuits), caches, RAM (Random Access Memory), ROM (Read-Only Memory), and other memory and storage devices including volatile and permanent storage used for transient and persistent files, buses and connectors, various transducers for local and remote communications and device interfacing, including radio frequency (RF), Infrared (IR), optical, fiber, coaxial cable (baseband or broadband), telephone cable, multiplexors/demultiplexors, and modems or other analog-to-digital converters, and direct connections to peripherals, including input/output devices, including displays, keyboards, and pointing devices, and to other equipment, including A/V equipment, including TV monitors,

speakers, microphones, and cameras. Elements supporting current TV/STB functions may further include, in the current OpenCable STB for example, tuners for in-band and out-of-band signals, NTSC and QAM demodulators, Point-of-deployment (POD) modules, MPEG-2 transport demuxes, MPEG-2 decoders and graphics overlay processors, AC-3 decoders and audio synthesis, NTSC encoders, IEEE 1394 interfaces and RF modulators, RF outputs, digital and baseband audio outputs, baseband, S-video, and component video outputs, and various digital outputs including game ports, data ports, and IR receivers and transmitters, as well as displays and keypads.

These systems also include software, including systems software, such as operating systems, network software, and middleware, and applications software. Such categories are suggestive and relative to the mission of the system. For example, browsers may be variously categorized as applications, middleware, or even operating system elements. Operating systems may be standard systems such as MICROSOFT Windows, UNIX, LINUX, and APPLE Mac OS X, or embedded operating systems such as MICROSOFT CE, PALM-OS, WINDRIVER VXWORKS, MICROWARE OS-9 and DAVID, as well as other system software such as Jini, JXTA, .NET, Web servers, Web services, agent systems, and programming languages and environments, such as JAVA, C, C++, C#, J2ME, JavaTV, Java Virtual Machines, FLASH, and the like. Standard file systems and database management systems such as relational (typically using SQL) or object or object-relational databases may also be employed, as well as alternative data structures such repositories and registries LDAP (Lightweight Directory Access Protocol), and storage structures, such as tuple spaces. The term "database" is used herein to refer collectively to all such collections of data. Application and middleware technologies might include those based on traditional "thick client" architectures that provide high function within a

user system, Web-like "thin client" architectures that rely heavily on browser functions and thus might limit local client logic and storage capabilities and be highly dependent on a server for richer function, and "rich client" architectures that might provide much of the power of a thick client, but might be capable of operating within a browser runtime environment and thus gain many of the deployment benefits of a thin client, or any combination or variation on these models.

System elements may preferably conform to formal or de-facto standards, such as OpenCable, Open Services Gateway Initiative (OSGi), Universal Plug and Play (UPnP), Home Audio/Video Interoperability (HAVi), Video Electronics Standards Association (VESA) Home Network group (VHN), Architecture for HomeGate (AHRG), AUTOHAN, MHP (Multimedia Home Platform), DASE (Digital TV Applications Software Environment), and the like. Digital Rights Management (DRM) technologies may be provided, including devices for decryption and for identification of users and their entitlements, including OpenCable Point-of-Deployment (POD) modules, smart cards, or others.

These devices, device sets, and systems are meant to be representative of the full range of current and future devices and configurations that may be suitable for use by a user or group of users to view hypermedia content such as ITV, whether in a home or office, or other context such as in a car or using wearable devices (such as head-mounted display, HMD), or immersive environments such as CAVE, or even implantable or bionic devices, which may include heads-up display, retinal projection, neural or EEG (electroencephalography) interfaces, and appropriate controls. Devices include the full range from conventional and digital TV and enhanced TV, PC-type devices, whether desktop or portable, personal digital assistants (PDAs) and cell phones. It is expected that there will be ongoing convergence among all sorts of devices

that allow access to and interaction with content, but that such devices will continue to group into families with different form factors and usage orientations. Major categories are likely to be as shown, with TVs being oriented to passive across the room viewing, primarily tuned to video, PCs oriented to active lean-forward use, primarily tuned to rich multimedia interaction, and PDAs for handheld use, with more limited screens and controls.

Content sources are intended to include all electronically accessible media, notably TV, movies, audio, multimedia, Web and other text, and online transaction systems. TV includes broadcast, satellite, cable, video-on-demand and pay-per-view, as well as stored content on varied storage media. Local storage includes hard disks, DVD, CD, VCR, TiVO/Replay, etc. Multimedia includes all forms of video and audio including hypermedia and virtual reality. Web and other computer content and transactions include all forms of Web content, wireless portal content, shopping and other transaction systems, text and multimedia databases and search systems, data processing and information systems, and the like.

Networks include direct connections between these elements, and various advanced network services, and these are essentially equivalent with regard to the intent of the coordination methods described. Major categories include home networks and LANs, whether wired or wireless (using such technologies as infrared (IRDA standard, etc.) and Radio Frequency (Bluetooth standard, 802.11X, etc.), the Internet, including the Web and streaming media and e-mail and other applications, and wireless networks including analog and digital telephony and access to Internet and other content and transactions, including access through portals using such technologies as WAP and iMode.

Turning now to FIG. 3, therein is depicted exemplary typical displays for ITV hyperbrowsing. These may include standard TV screens and standard computer and PDA screens, with a wide variety of combination cases, and with variations as to form factor both for the display and the input controls and devices. A simple example is basic TV/video screen 310, depicted as presenting a video program "A.". This is just a standard video image as normally presented directly onto a TV monitor, of whatever resolution, whether standard definition or high definition, or otherwise. Optional variations relevant to ITV systems include the overlay on the main screen of a simple graphic, sprite, or bug 312, shown here a an "*i*" like the bug used by the WINK ITV system, that is displayed when ITV content is available for the video segment currently showing. Other simple variations include addition of simple overlay area 314, representative of various similar overlays that can be used to present text or other information (which may cover a portion of the TV image, or cause the TV image to be shrunk. Similar overlays may be inserted into a video signal at a distribution source, as is now common for news and sports programming, such as on FOX, but with digital STBs, such overlays may be inserted by the STB or ITV system at the user site.

A more advanced ITV screen typical on what may be produced by a common ITV system driving a TV is shown as ITV screen 320. This represents an active navigation of ITV hypermedia or hypermedia-like resources, including menu 322, which provides a simple list of options, usually in simple text, but potentially with graphics as well. In a typical navigation process from the basic program screen 310, such a screen may be obtained as a result of entering a select key on the remote control, but it may come from any interactive step, using any of a variety of navigation controls. On selection of an entry from the menu, interactive content screen 324 may be presented. Depending on system design, this may fill the screen, or appear

with the menu 322, or may include a further menu, not shown. A further feature shown is picture-in-picture (PIP) frame 326, which is a region of the screen used to present a reduced scale video image. This may be the base program to which the ITV enhancements relate, or some other video resource. Alternatively, such a video frame may revert to full screen, and exclude other items from view.

The comparable, but much richer ITV screen typical of a more high-resolution computer based ITV or similar hypermedia browsing experience is shown as PC ITV window 330. This basically has all of the function typical of GUI displays. Typical features include menu bar 332 with active menu drop-down list 333 that responds to a user selection without need to change other screen areas, interactive content 334, which may be a Web page or other format, video window 336, which may be placed in a variety of fixed positions or positioned by the user, or embedded in interactive content page 334, and task bar 338, which can be used to switch among other active windows. Window 330 may appear in a full screen maximized view or an intermediate size (as shown), or may be minimized to be hidden except for the appearance of a tab on the task bar 338, which can be clicked to bring back display of the window. Additional windows may appear concurrently, in various configurations, such as stacked as shown, tiled into a mosaic of frames, and the like, or remain hidden in virtual layers.

Also comparable, but simpler, is an example of a relatively constrained PDA/phone screen 340. Here activity typically results in replacement of one screen with another, sometimes with limited combinations on one screen, as shown for menu screen 342, content screen 344, and video screen 346, shown with a small menu area included. Use of video on PDAs and phones is not yet common, but is expected to become so.

It should be understood that in addition to conventional GUIs and the basic UIs specifically addressed herein, these methods are also directly applicable to other UI approaches. These include advanced interfaces that go beyond the traditional GUI of the WIMP (Windows, Icons, Menus and Pointing Device) paradigm, including LifeStreams, data walls, and richer 3DVR, collaborative virtual environments (CVEs), and multi-sensory-modal UIs. It will also be understood that while these display variations have been described in connection with their traditionally usual hardware context, that connection is not essential, and that there may be uses with other hardware configurations, such as, for example TVs that support high resolution GUIs (such as on high definition monitors), PCs that present low resolution ITV-style UIs (such as for lean-back use), and other variations.

One simple UI variation that deserves mention for its common use in current TV/STB-based ITV systems is the use of alternative navigation methods to obtain interactive services that are not directly coupled to a base TV program, but may be obtained by entering a special channel number or by selection from an EPG. Variations on these methods have been called virtual channels or walled gardens or portals (and these may also be described as asynchronous). Like Web portals or walled garden services, these may provide some selection of services, such as weather, news, sports, shopping, and the like, that are available on demand. It will be apparent that, for purposes of the methods described herein, selection of such a virtual channel directly, via an EPG selection, or by any other method is just another navigation action, and that the session transfer methods described herein can be applied to any such navigation action, using similar command variations to specify the targeting of a transfer request.

Referring now to FIG. 4, therein is depicted an exemplary schematic of state data relating to two systems and in a migration process. Depicted is a base state 410 for a system A, and a

base state for a system B, where each system has multiple browser sessions, A1, A2, B1, B2, and the like, each of which may constitute a distinct user session. A complex user session may actually involve active use of multiple software tasks, each constituting an application session running different software applications, such as writing with a word processor and referring to Web references, but this discussion addresses the case of sessions based on browsing tasks. This is further simplified by considering user sessions on different browser instances at a given device set as separate browser sessions, so a user session may have multiple browser sessions. Some discussion is given to migration of such compound sessions combining multiple browser instances at a given device set, but for simplicity, much of the detail focuses on the case of migrating a single browser session. Based on these teachings, extension to the multi-session case will be straightforward for one skilled in the art.

One aspect of the present invention is the abstraction, extraction, and exchange of session state data that specifies the current state of a live interactive session in progress. The base state 410 of a given system A includes static settings 412, which control user options as to how the system behaves and presents itself. These include image brightness and contrast settings for a display and a wide range of preference settings for a PC or other system and its suite of associated software applications, again with emphasis here on the browser.

Complementing this is the transient state data 414 that defines the current status of an interactive user session relating to browser session A1, and similar transient state data 416 for session A2. A system may have a number of sessions in progress at once, whether independent or related, each defined by the state of navigational position through multimedia content (including the time-position in continuous media content and the special position in spatially-oriented content), the contents of various input and output elements and controls, the nature and

configuration of open windows, menus, drop downs, text entry boxes, check-boxes, etc., as well as the current state of work files, buffers, databases, logs, in-flight transactions, embedded logic objects helper applications (such as streaming media players), etc. The transient state data for a browser includes the identity (URL) of the resource being viewed, and at the time of link traversal, includes all current state on the link arc and the process of traversal. Depending on the nature and state of the session, and on the type of systems being used, some or all of this transient state data may be needed to migrate a session from one system to another.

Supplement data 418 for system A, not normally specified explicitly, can be formalized to further describe the characteristics of that system A (such as coding conventions and other basic architectural attributes) that may need to be known to embody a corresponding session on a dissimilar system B.

The portable state 430 defines the subset of all such static and session state data that may be needed to migrate any or all selected sessions from a system A to any other supported system B, and the superset of data needed to migrate a single specific task. This may exclude some local state data in each portion of the base state that is not relevant to re-establishing a session in useful form, and at the desired granularity, on another system.

Given a request to transfer one or more sessions from A to a specific system B, a transfer state record 450, shown here for session A1, can be assembled. This includes only the relevant data to the specified sessions, and only the subset of that state data that is relevant to the capabilities of system B. This portable state information can then be used to add an equivalent session A1' to system B. This is shown in the schematic of the base state after migration 460 for system B. Depending on the nature of the request, this equivalent session may present the current resource presented for A1, or the new ending resource resulting from a link traversal

initiated from A1. Also depending on the nature of the request and relevant user preference settings, static settings data from A might or might not be relevant to the transferred session A1'.

Turning now to FIG. 5, therein is depicted exemplary further details of a migration as it is effected. System A 510 is shown as including session A1 520, a set of user interface devices and controls 530, and a state importer/exporter/tracker 540, which may be implemented as a module that can be used with a standard browser. Initiation of a migration request results in the creation of transfer state record 550. (The term state set may be used synonymously with state record.) This transfer state record is used when a migration is triggered, which may occur in multiple ways. One way is that a user at one system requests that one or more sessions be migrated or transferred from a system A to a system B. For the example shown, a user at system A interacts through available user interface (UI) controls to conduct a session, and then makes a request to migrate a session to system B. State importer/exporter/tracker service component 540 provides these services. In the example of a user at system A requesting that session A1 swing to system B, this request is processed by the exporter services on system A 540, which extracts the portable state, creates a transfer state record 550, and passes it to system B 560. The corresponding importer services 590 on System B then use that data to activate an equivalent session in-progress, A1', on system B. Alternatively, the migration may be triggered by other means, for example based on coding of target attributes for the link. Processing of such cases is essentially the same.

Optional features may allow the user to specify any of a range of cases for coordinating ongoing activity on the two parallel sessions A1 on system A and A1' on system B. These may include terminating the original A1 on system A or leaving it unchanged. In the case of migrations from a base TV program, leaving A unchanged might typically be preferable, but for

migrations from interaction with currently-displayed enhancements for an ITV program on a TV, terminating the enhancement session may be preferable. For the case of migration from a PC, leaving the session unchanged may be preferable. An optional capability provides for ongoing interaction with the two sessions as one linked, shared session on both systems, acting like a collaboration system (or a fully synchronized multimodal session). Variations could make the session viewable on both systems, and could permit either or both systems to interact with and control the ongoing activity of the session. This is shown in FIG. 5 as additional transfers of state, with #1 creating the session on system B, as already described, #2 relaying an interaction on system B back to system A, and #3 relaying a subsequent interaction on system A back to system B. Such relays of ongoing interactions can be conveyed by transfer state records such as the session A1' state record 555 shown for #2. Thus users could treat the sessions A1 and A1' as a single logical shared session, in a manner similar to that used in conventional collaboration systems, in which all significant interaction events are replicated and synchronized as they occur. Such systems might add the features described here for non-synchronous migrations as an added feature.

Referring now to FIG. 6, therein is depicted the flow of an exemplary process 600 of transfer showing export and import of state. The process begins on system A with an interactive session in progress (step 605), in this case a browser session A1. A transfer request to transfer the browser session A1 to system B is initiated by reception of some trigger event (step 610). Typical trigger events include a user request to re-target an ending resource or to duplicate the current resource to the target system, whether to a new window or an existing window. Alternative events include link attribute coding as specified by a target attribute (following the model HTML) or a show attribute (following the XLink model) which may be triggered on link

activation, or on load, as specified by an actuate attribute. To prepare for that, a transfer state record is assembled by exporter/importer/tracker for browser session A1 (step 615). With all necessary information on the session to be transferred assembled and packaged for transfer, that state record with any associated information is exported to system B (step 620). This may be done by direct communication to system B, or via some intermediary controller system. Depending on options selected, the session A1 at system A may be terminated, left in as is to continue independent of the transferred version A1' running on system B, or, if collaboration/synchronization features are supported, tracking may be applied to keep the two sessions synchronized as events occur on either or both of system A and B. In the case of such tracking, the exporter/importer/tracker on system A exports similar state records (or simplified event records) to echo all relevant interaction events to system B (and imports any corresponding events from system B, as noted below) (step 625).

Meanwhile, at system B, unrelated activities are presumably in progress (step 650). Alternatively, system B could be idle, or it could be off, and might preferably have support features to sense and activate it on receipt of a transfer request. On receiving the transfer request (from step 620 on system A), the exporter/importer/tracker on system B imports the state record for session A1, interpreting and converting details as needed to accommodate any differences in capabilities, architecture, and preferences at system B (step 655). The exporter/importer/tracker then sets up session A1' as an active browser session on system B, loading the desired resource, and setting up other aspects of context as appropriate in accord with the transfer state record (step 660). If the target is to an existing browser session, this setup activity can be limited to making just the changes resulting from the transfer. Optionally, if collaboration/synchronization tracking was requested, further steps by exporter/importer/tracker on system B will maintain

tracking to echo all relevant interaction events in either direction (step 665). Such tracking may optionally involve any number of additional systems as well (with additional transfers to first set them up, as well).

As noted, embodiments may add the MMUI support methods just described to existing browsers, using methods that will be apparent to those skilled in the art. In some cases this may be done in the form of external modifications that could be done by third parties, and that may be retrofitted to installed systems. For example, with MSIE (version 4.0 and later), there is formal support for Browser Extensions, relying on APIs that provide access to browser functions, including MSHTML, the WebBrowser Control, and the associated objects, interfaces, functions, enumerations, and hosting features and ActiveX Hyperlinking and Travel Log and other features that is extensively documented on the Microsoft Developer Network site (including Programming and Reusing the Browser, WebBrowser Customization, and related sections). Similar capabilities may also be available for other platforms, including PDAs, tablets, and STBs. Alternatively, new versions of browsers may be created to add this functionality.

As noted, state records can be transferred directly between coordinating systems, or via intermediary controller systems. Other variations may also be useful, such as using special state intermediary repositories or databases. Standard interchange structures with well-defined formats and based on standard interchange frameworks or metalanguages such as XML may be desirable to facilitate interchange of such state details among systems that may have heterogeneous architectures and may use different browsers (or editors). One method is to communicate state via a tuple space. Linda-like tuple spaces offer attractive properties as a state/event exchange medium for coordination systems in general, because of their flexibility and associative properties, and their application to the methods described herein will be apparent to

one skilled in the art. Relevant developments include implementations of programmable XML dataspaces that support distributed, federated tuple spaces and that add reactive properties for more flexible dynamic and rule-based behavior, and related work such as that described by Cabri, XML Dataspaces for Mobile Agent Coordination, SAC'00.

At a broad level, the transfer process described herein may appear to be similar to software task migration, in that a task at one system is migrated to a second system. However, the transfer process described herein differs for at least the reason that it is the session state of the task -- not the actual program performing the task -- that is migrated.

According to certain embodiments of the present invention, the transfer may draw on application awareness of transfer functionality to facilitate export and import of state. It will be understood by those skilled in the art that these methods of transfer-aware application support, relating to export and import of state, could be applied not only to browsing applications, but to most other kinds of applications as well, including for example, word processing, spreadsheet, analysis, drawing, database management, transaction processing, and the like. Adaptation of these migration or transfer methods to such other applications would primarily involve adaptation to the particular elements and granularities of state relevant to the particular application and need.

While adding migration-aware functionality might impose a development cost on each application, these methods could be simpler, more efficient, and more readily achievable than more general application-transparent methods, including those that might operate primarily at an operating system or programming environment level. Alternately, perhaps to provide "transparency" to applications with regard to transfers, application awareness of transfer functionality might not be drawn upon.

These methods have been described for the case of a hypermedia system architecture and coding conventions similar to those now in use on the Internet (with HTML, XHTML, and XLink) and in current ITV systems, but they are equally applicable to alternative embodiments, as will be clear to one skilled in the art. Current Internet and ITV systems generally are based on outbound links that are contained within a starting resource, but for some types of resource, such as image or video, the link may not be in the resource itself, and thus technically a third-party arc with respect to that resource, but is directly associated with it, being contained within the context in which the resource is distributed, such as the Web page that loads an image or video, or the TV channel that includes a TV program and has associated enhancement, such as in the VBI (Vertical Blanking Interval, such as NTSC Line 21 or PAL line 22) or VANC (Vertical Ancillary Data) or in the MPEG stream containing the TV program, possibly using SMIL, or in some other stream from the same feed source, such as a data or object carousel, or the like. Other past and possible future hypermedia systems provide richer linking methods, including richer use of third-party arcs, and the methods described above are equally applicable to such architectures.

Basic Device Set Management and Communications

As foundation support to the browsing process just described, it may be desirable that a device set management process that performs basic setup and update functions be applied to pre-identify and dynamically discover device sets that may be used in coordination with any given system, to define combinations of such device sets as composing designated device set groups, and to set preferences for use of device set groups and device sets within groups. This provides a framework for determining what transfer options should be considered and taken under specified conditions.

This communications process could desirably be based on and compatible with related lower-level processes and standards defined for linking such existing devices and systems, such as UPnP, HAVi, and OSGi. Methods such as defined in those standards could be applied to enable basic communications among the devices, to provide discovery, presence, registration, and naming services to recognize and identify devices as they become available to participate in a network, and to characterize their capabilities. Useful communications services could include messaging services for basic coordination, event services that can be used for tracking events, as well as streaming services that can be used for relaying signals from one device to another. These lower level network services provide a convenient base for the middleware and/or application level coordination services described herein. Naturally these higher level services can be provided independently of such standards or any available software that supports them, to the extent needed to support desired devices and services, but use of available software and services could simplify implementation and have numerous well-known benefits associated with use of standards and COTS (commercial off-the-shelf) devices and software. (Note that HAVi uses the term "target" in connection with remote control action and observation commands to refer to hardware devices, not applications, and this usage differs from the hypermedia application-related notion of link targeting addressed herein.)

It may be desirable that communications among local systems and devices be done using local network facilities, such as a LAN or HAN or the like, or direct local connection, and that wide area networking to other locations be used primarily for access to external resources and services. However, limitations in available facilities and support may make it necessary at times for such communications to be via WAN, as well, even though this may be counterintuitive. This is likely in near term embodiments, such as with STBs that have communications to head-

end cable or satellite systems (and through them to the Internet), but not directly to local PCs or other systems. Legacy STBs may have no external local communications capability suitable for such use, and even those that do may not be commonly connected to the same network as the PC. PCs and the like may in many cases connect to the Internet via dial-up or DSL or other facilities unrelated to those used for the STB. In such a case, the more circuitous external path may be quite serviceable for the coordination tasks described herein. Such a path could be a pure Internet path, such as using DOCSIS support from the STB to a separate cable modem that connects to the PC, and which merely routes through the head-end, or it could use other protocols from the STB that may require conversion and relay to the Internet by a server at the head-end.

This method of relay via a wide area network can be broadly useful, including for cases that do not involve cable TV services and STBs, but might employ other network and device technologies. Such use of a WAN could substitute for a local connection between any device sets to be coordinated. For example, a PC or PC-DTV system (or a TV/STB), or other device set might obtain resources via IP or other protocol over any satellite, wireless, DSL, fiber, or other transmission path (or locally) and could coordinate in a similar manner with an independent device set (that also has remote communications facilities) over that or any other bi-directional wide-area path. Such linkage could be on a direct peer-to-peer basis or be mediated by a server (whether remote or local). As a further example, a DVR, home media server, advanced TV, PC-DTV or the like might obtain streamed video and movies from an Internet service such as that of REALNETWORKS, and could coordinate a session relating to those resources with related activity, such as an enhancement session, at another device set, through the Internet. Such coordination might involve direct transfers to and from the second device set over the Internet,

using the methods described above, or use relay through a remote server, such as one that might be provided by REALNETWORKS or others. As will be apparent to one skilled in the art, such relay via a remote server, and possible provision of related value-added services, could be done in a manner substantially equivalent to that discussed further below with regard to cable head-end servers.

The hardware context that has been described may be impacted by the emergence of "modular" computer systems in which a core computer module may be swapped in and out of multiple sleeves, carriers, docking stations, or other connection matrices and used in conjunction with different user interface I/O device sets of varying form factors, such as desktops, notebooks, tablets, and PDAs.

For such hardware devices, by providing for hot swapping such that the transient I/O state of a session could be reestablished with a swapped device set, an effect having some similarity to a session transfer could be achieved by physically moving the core module from one device set to another. .

According to embodiments of the invention, such hot swapping capability could be provided by adapting the export/import functions such that a transfer was done, not by transferring the state information to another processor, but by recognizing the change of I/O device sets connected to the single core processor and reapplying the resource rendering and adaptation functions to take into account changes in resolution and related UI style adaptation, as described further below. This would effectively substitute the renderings on one set of devices with the equivalent set of renderings on the new devices. In addition to this cloned resource case, transfers could also involve a link traversal, with a change from a starting resource on a

first connected device set to a selected ending resource on the new device set, but the cloned case would be simpler and perhaps more generally useful.

However, much like a brain transplant, such a core module transfer would disconnect the original device sets and presumably move or halt all sessions controlled by that core module. Accordingly, such an implementation would not address, for example, the general objective of MMUI use relating to the ability to use the multiple device sets at will, such as in a multitasking situation in which each of multiple sub-sessions may be concurrently active on different ones of the multiple device sets. For that kind of use, multiple processors would still be needed, with transfers accomplished as described above, regardless of whether the processors were modular or not.

Push and Pull Methods For Controlling Transfers At Either Source or Target Device Set

Embodiments of these methods can allow that transfers be triggered from either the source or the target device set. One case, as described above, is a "push" trigger that is activated while browsing on the originating (source) device set. The alternative case, which can be provided as a complementary feature, could permit the user to act from the target device to trigger a "pull." In this case, instead of system B waiting to receive a transfer request, a command on the as-yet-uncoordinated PC could actively request the transfer from the TV, signaling to the TV's exporter to send a state record back. In a basic embodiment, this could be as simple as a request to the TV for its current channel, which can be accommodated with any TV that responds to basic commands such as might be provided with network support (such as HAVi). This could enable a simple user command to "present enhancements to the channel I am now watching." The state record can also include time-position information and more specific

address details on the current program resource to accommodate various kinds of programs, including stored resources, video-on-demand, streamed content, advanced feature states such as camera angles, second audio program, closed-caption, and the like. Pull transfers can also be supported during fully interactive sessions (on all kinds of device sets), including the same functions as described for push transfers, and with the same flexibility in specifying whether the original session is to be terminated or left as is (or put into collaborative synchronization).

Pull transfers might add some additional complications, in that such transfers might be supported only at appropriate breakpoints in browsing activity, and only with appropriate permission. An enabling control could be provided as part of the exporter to work with the browser to ensure proper function, consistent with the granularity of state transfer supported. This might involve refusing some pull requests, queuing some requests to be held until a suitable activity breakpoint is reached, or satisfying some requests as of a recent prior breakpoint state. For example, during a forms entry process, a pull request might be refused, held until the form is submitted, or accommodated with the state set back to the initial state of the form. Security methods might presumably be desirable to verify that push and pull requests are permitted on both sides, as noted below.

Resource Access

With regard to the underlying hypermedia resources, it may be desirable that these methods rely on the device sets making access to the same resources from the storage layer. Given that heterogeneous device sets such as TVs, PCs, PDAs, and the like require significantly different presentation styles, this is may be accommodated by adaptation at the client, and this may be based in part on alternative style recommendations contained within the resources. Thus,

for example, an ITV enhancement resource (such as a Web page) could be coded to indicate one style of presentation for a TV screen, perhaps the same or slightly richer presentation for a PDA, and a significantly richer presentation for a PC. As has been recognized for the related case of phone and PDA access to the Web, this reliance on a single source with multiple style codings, such as using XHTML or CSS, offers significant advantages in content management and flexibility, and this has recently led to broader attention to device independence within the W3C. Useful methods might be drawn from the ongoing work of the MPEG-21 Digital Item Adaptation effort, as will be apparent to one skilled in the art.

This distributed, specialized, ad hoc approach to rendering also simplifies the demands on the browser, and facilitates optimal control of rendering appropriate to any device. Each system can be equipped with a browser specifically suited to the rendering tasks appropriate to the device sets it normally supports, avoiding the need for a common super-browser able to support any device that may be joined into a MMUI. Similarly, it avoids the need for coordinating alternative browser rendering processes dynamically, since binding is to the common stored form of the resource. Details of varying adaptations can be left to the target browser and need not be resolved until presentation time.

Other efficiencies may also be achieved by direct access. One relates to bandwidth efficiency, since stored forms of resources are generally designed to be compact and bandwidth efficient for the particular kind of content involved, while alternative methods of transmitting data in a display image or other partially rendered form may be less efficient. Another relates to the complexity of a two level relay and conversion with an intermediate system, as described previously with regard to the WTS architecture. A further example of benefits of direct access in the case of video is in the use of receiver-driven layered multicasting (RLM), which layers

video into multiple multicast streams so that a receiver subscribes to only the streams necessary to get a desired resolution. Since a PC device using video as secondary content, or possibly for pointing device/screen support secondary to the TV, could need less resolution than a TV or HDTV, the PC can obtain the video at this reduced, less intensive level. This may obviate receiving the additional layers at all, or at least eliminate them from being forwarded from a TV system to the PC device set (in whatever format).

This single-source approach may be valuable in simplifying the task of ITV deployment, by cleanly decoupling content issues from presentation system implementation details, except for cleanly specified style variations. Content producers need be less concerned about which architectures and form factors are being used for viewing, and need not face fragmentation of their markets caused by incompatibilities in viewing system. At an initial, base level implementation, all enhancements could be coded for a TV form factor viewing alone, and such resources could be usable, if not optimal, on PC form factors and on most tablet and PDA form factors. Thus content providers can have full reach to all form factors, and can selectively add style variations to those resources and for those form factors that warrant the investment. Techniques for automated style transformation (based on set rules and style sheets and/or more adaptive programmed transformation methods) could also be applicable with more or less workable results. This is similar to the conversion now done in some cases for phone access to Web pages, but it can be expected that up-conversion from low to high resolution, as desired here, could be much more effective than down-conversion, as done for Web-phones. A promising short-term method is to create server-based adapters (or proxy servers) do this up-conversion and concurrently adapt access and coding from native proprietary ITV formats to Web format. These services could be architected much like "clipping servers" used for down-

conversion of Web pages for access by phones and PDAs, but instead performing the up-conversion to Web pages in what might be called a "composition server" that combines small pages and short, simple menu controls into larger, richer pages with more powerful and varied controls. Over time, a preferable method may be to adapt current ITV content and presentation systems to use Web technologies and standards (such as HTML and HTTP, or newer standards such as XHTML) as native formats for resource access and coding, still providing for the small pages and simple menu controls suited to TV form factors as at least one of the included styles (but gradually adding the improved capabilities to finely control alternate style codings for richer form factors, as described). Advanced standards for style specification and transformation such as CSS (Cascading Style Sheets), XSL (Extensible Stylesheet Language), XSLT (XSL Transformations), or RML (Relational Markup Language) can also be applied, as can the use of embedded programming objects such as ASP (Active Server Pages), JSP (Java Server pages), or the APACHE Struts Framework. As will be apparent to one skilled in the art, any of these current and emerging methods can be used to give the effect of variant resources for each of a number or form factors. For example, use of RML or XSLT could permit page templates coded in HTML, XML, XHTML or other formats to be transcoded (to or from one another) based on the structure of the content and the context of the target device set to allow for changes in UI elements, pagination with automated insertion or collapse of links, and other styling actions, and related methods can be used to control the styling of link presentation and traversal as well.

It should be noted that in embodiments where direct common resource access is not practical, it may be necessary for link arc and/or resource data to be transferred as part of the state information (or as a supplementary element or stream of elements). An example of such information might be ATVEF triggers and embedded resources.

Linkage and Transfer From Pure Video Sessions To An Enhancement

The case of transfer from an interactive resource is generally as described above, but the case of transfer from pure video is in some ways simpler, with some aspects that may be further clarified. For simple video there is less context or state information needed, with one element being the identity of the program and another element being the time position within the program. In certain cases there may also be spatial selection information, such as when activating a "hot spot" as a starting resource that indicates a specific region, corresponding to some viewed object (such as an actor or player, or an item of merchandise). Some or all of this basic information is readily available from any advanced TV system or STB, and can be expected to become universal as such systems and associated home networks proliferate. It may also be obtained by external means, even with basic TV devices, as noted below. (Certain cases may also involve active enhancement resources, as well.

According to embodiments of the invention, if the video is from a realtime broadcast (or other realtime source, such as a camera), the time position can be taken to be just the current real time. Should it not be realtime video, or should it be desired that it not be treated as realtime, time-position information (relative to the beginning or some other reference point) can be obtained by a number of methods, including reading embedded time code data, or externally tracking time-position. The TV exporter can extract or derive such time code data to include in the state record. If VCR-like trick play functions or other hypermedia controls are allowed to alter the play of the base program, and synchronization of enhancements at another device set is to be maintained, then a tracking process can be used to advise the other device set of the resulting changes in state (time-position) as they occur.

Program state information may also be available from an intelligent remote control, which will ordinarily have information on the channel or other program setting last set, and which may also have time-position information, or be able to construct such information based on analysis of the commands it issued (as long as there are not intervening control inputs from another source that the remote is unaware of). From this perspective, it should be understood that to the extent it is an intelligent command device, such a smart remote can be considered the controlling processor for a TV (in parallel with the STB or other control system) for purposes of coordinating activity with another system, with or without special support from the STB. In the case that an intelligent remote control includes more robust viewing state awareness, such as in the case of a remote control that provides EPG access and viewing control, its ability to serve as a source of state information is enhanced. Other external devices may also be used to sense and transfer the TV state, such as a device that monitors IR signals from the remote control, or that senses channel indicators on the TV set or coded into a video image. One method of providing coding of program identifiers or links or other such data within a TV program resource in a form that can be extracted externally even from a standard TV set with no special signaling support, is to insert video-image-based or audio encoding into the TV program. Such codes may be directly understandable by the local devices as a program or resource identifier or as link arc information or the like, or may be relayed to a remote service for interpretation. Other alternative sources of such state information may be DVRs and similar devices, as discussed below.

In addition to identifying state, TV systems also may present challenges in identifying link arcs. As just noted, video may be treated as containing links, or as using third-party arcs. In the case of simple video, third-party arcs may be obtained in the form of a linkbase, from a TV feed-related source, from the Web or elsewhere, or may be derived by some other link-like

process that leads to a source of associated material that may be synchronized with the TV program or not.

More advanced systems, such as those employing ATVEF/DASE/DDE, DVB-MHP or similar methods, may embed link arc information into a TV stream, such as using VBI or MPEG, or into related channels such as a DSM-CC, ATSC or DVB-MHP data or object carrousel. In the case of ATVEF, for example, triggers are embedded into the TV stream as real-time events (called broadcast triggers) that employ the current stream as starting resource and may contain a URL for an ending resource, along with a human-readable name, an expiration date, and a script, which is to be executed by a trigger receiver object within the ending resource, either automatically or after some user selection. Receiver Web pages containing trigger receiver objects are expected to receive and process the script to cause the desired presentation action. ATVEF also provides conventions for a Local Identifier URL Scheme (LID) which serves as a URL relative to a given namespace that can be local to a distribution channel that can be used for resource delivery and that may be apart from the Internet, such as by broadcast, cable, or satellite. ATVEF also provides for bindings to the particular channel standards that are to be used, for session announcements in accord with SAP and SDP, which may include multicast, and for a Unidirectional HTTP (UHTTP) protocol that adapts HTTP to one-way channels (with provision for separate back-channels). ATVEF Transport A provides for support of a data return path or back-channel, while Transport B is for one-way broadcast. One problem with ATVEF and similar real-time linking/triggering schemes is that they may have difficulty in adapting to storage and replay of content time-shifted to a time other than the original broadcast. Another problem is that the trigger information, as well as the associated enhancement resources, may be inaccessible to an independent system that is to be coordinated, unless it has its own duplicate

TV tuner, STB, and entitlements (even if such access is to be used only in coordination with a TV for a single viewing).

Current 2-box "Enhanced TV" offerings (such as, for example, those from the ABC and PBS networks) may be problematic in that they require the user to know the identity of the current program and to know how to locate a corresponding resource, which is a Web page for that program, and manually navigate to it, after which the Web enhancements can be navigated with purely temporal synchronization maintained from the Web site. For example, for the show "Who Wants to be a Millionaire," a user must go to www.abc.com, click a button marked ETV, then pick Millionaire (and then select the time zone). For a similar service produced by ACTV for MTV2, a user must go to www.mtv2.com. Specific URLs may be directly accessible to a TV/STB, or to a PC containing a TV tuner, which could receive ATVEF or similar triggers, but such link arc information is not currently accessible to a separate PC lacking a TV tuner (and a separate PC with a tuner could still need to be manually tuned to the proper channel). This is obviously awkward and burdensome to the user, and limits the ability of such offerings to gain wide use. It is further complicated by the fact that the user may be unaware of the existence of such a page, and that there is no consistent pattern for how different program sources make such pages available—there are no navigation conventions that are common to many programs. It is evident from the teachings provided herein that this should preferably be supported as an automated link traversal that could relieve the user of these tasks and the need for related navigation knowledge.

An alternative embodiment could make use of a table of associations that could serve as a simple linkbase to allow automated traversal from the TV program to the Web-based enhancements. Such a linkbase could be pre-defined by the user, or preferably obtained from the

Internet or some other service. Still other embodiments might obtain such linkbases from other sources, such as embedded in the TV distribution feed, much as for ATVEF. This method can avoid the problems of need for access to the TV signal to obtain link arcs, and of time-shifted viewing noted above.

In particular, such a linkbase could be structured similarly to an EPG. This could exploit the simple basic structure of the EPG as being a two-dimensional data array, with one dimension being time and one being channels, that is readily searched by those two keys. To use such a structure for a linkbase is just a matter of logically placing the link arcs for a given time and channel in the corresponding grid slot. This could be done in a separate EPG-like structure, or integrated into an EPG and possibly distributed by the same services that provide EPGs. Multiple arcs could appear within a given grid slot, with starting resource information that narrows them to specific sub-intervals or to specific time/region portions of the resource, and with other attributes that permit selection of alternative links based on defined parameters and filters, such as to personalize links based on a user profile, or to give users a choice. This method of providing a collection of link arcs having time-interval attributes (start and end times for enabling the link, relative to time position in the starting resource) serves as a more flexible alternative to the sending of link arc triggers in real-time. The effect of real-time triggers can then be derived by processing the linkbase in a sequence that is in accord with the time-interval attributes.

Such EPG-like linkbase structures can be composed at multiple levels, one at the program and channel level, with multiple alternative links as just described, one at the feed source level, including all channels from a given distribution source, such as a cable or satellite system or an Internet streaming service, and one aggregate level that combines multiple such distribution

sources into a broader suite. Such linkbase portions may be acquired pre-assembled, much as for an EPG, or some may be dynamically constructed from various sources and with updates in real time. In certain embodiments, a linkbase manager program might be used to assemble and maintain this linkbase structure, in conjunction with one or more external linkbase supply services.

To better accommodate programs that are not appointment viewing and synchronized to a given real time, but can be obtained on demand in some recorded form at flexible times, an alternative to the time-based EPG structure is a one-dimensional table keyed on program identifier or resource address (such as names, URLs, URIs, and the like). As DVRs and similar devices proliferate, most or all TV programming may tend to fall into this category after its first availability. This may be moderated by the use of time-phased release windows, such as currently used with movies, that could restrict viewing, copying, and VOD access for some initial periods. In any case, such a structure can be used in combination with the EPG-like version to provide a flexible combined linkbase system that can be searched either by a definitive and unique program identifier, or by a time and channel combination.

It should be noted that current methods of embedding links into TV program transmissions (such as ATVEF) can be limiting, and that provision of linkbase metadata in a format that is separable from the associated TV/video program mediadata encoding can facilitate more flexible use of the linkbase, such as to facilitate use with stored programs. Whether provided as a complete linkbase, or locally assembled during reception of a linkbase stream, such a table need only be archived with an appropriate identifier and made available at playback time. The table entries can be used by a clock-based event driver to reconstruct a realtime trigger

stream on demand, either from a remote server or from a local application (such as, for example, using a browser accessory).

Also, reflecting the wide range of possible linkbase embodiments, it is helpful to think of linkbases as possibly being virtual, in that they may not be physically assembled into a single data structure of the sort just described, but that the effect of such a structure is obtained by some process of finding and using link arc data based on time and channel identifiers, based on unique program resource identifiers, based on link arcs being supplied with the program, or any other suitable process, and that certain embodiments could work with any suitable form of virtual linkbase and any suitable data structure. Such a virtual linkbase might also be embodied purely as a process, such as in the form of a resolution service that resolves program identifiers into link arcs (or linkbases), acting much like a name resolver that given a starting resource identifier returns one or more ending resources (along with other link arc supporting information).

It should also be understood that multiple alternative sources of linkbases relating to a single base program may be accessible, possibly from multiple providers, and that viewers might be given controls to determine which one or more linkbases are to be applied at any given moment or time-span or anchor position, with what priority among selected alternatives, including controls for passive (automated and implicit) or active (explicit by the user) selection of alternative link arcs at the time of link traversal, as well as preference-setting controls to pre-set such choices. Such alternative linkbases may be organized into linkbase channels, and the controls might operate as "linkbase channel" and/or "enhancement channel" selector controls that operate much like a secondary selection to the conventional program channel selector control. Thus the user might first select a program channel, and then select one or more linkbase channels to be applied. It should be understood that transfer requests could be structured to

provide for one set of linkbases to be active at one device set, with another set of linkbases active at a different device set, with the linkbase selection being included as part of the state set. This could further exploit the power of MMUIs to allocate interaction with different sets of linkbases to different device sets, such as to allocate tightly coupled and non-intensive linkbase channels to the TV and loosely coupled but perhaps more intensive channels to a PC. Access, selection, and resolution of alternative linkbases may be facilitated by servers acting as linkbase proxy servers. Specific methods will be apparent to those skilled in the art based on the teachings provided herein, such as using methods similar to those described by Page, et. al., Its About Time: Link Streams as Continuous Metadata, at Hypertext '01.

Thus a transfer from the TV could assume that the PC needs only basic TV state as a starting resource to identify a third-party arc and establish a "transferred" browsing session with the corresponding ending resource. That ending resource might be a resource (such as a Web page) that is generic to the program (whether the specific program episode, or all episodes of a series), or one specific to the current time span within a program, or more specifically to a current time and an indicated image region. In the case that the arcs are embedded in the TV feed, those arcs may be interpreted at the TV, and could be followed there to view enhancements there, or could be passed to the PC or other device set as part of the state at the time of transfer. Alternatively, the target system could have access to the same links or linkbases, and enhancement resources, either directly, or via the TV system.

As use of DVDs, DVRs, Internet streamed video, or other similar alternative video sources becomes common, the DVD, DVR or computer or other controlling resource access device may sometimes serve as an alternate device to the TV/STB in controlling what is viewed on the TV. In such cases, the session transfer activity might originate from that device, rather

than the TV/STB (or in combination with it), and the methods described herein in the context of control by a STB should be understood as applicable to DVDs, DVRs or other similar devices as well. This may facilitate implementation of coordinated services for those cases, since providers of those devices may be more open to addition of the coordination support software, or the user may have direct ability to add such software. In the case of a DVR, for example, the DVR could create and forward the state record as described above, activated by either a push or pull request. DVD and DVR function and Internet stream access can be readily provided on standard PC systems that offer considerable extensibility. That also means that centralized, single processor/single system embodiments, as described below, may also be readily applicable in those cases, although in many cases it may be preferable that two separate systems be dedicated to video and PC functions. It is further noted that with DVRs, a common usage mode is to have even live programming (such as from cable, satellite, or broadcast) be obtained through the DVR (to enable pause, and other special features), and such practice may extend to Internet-sourced programs as well (and DVRs can include DVD players), so that coordination from a DVR may be applicable to most or all content viewed. The above discussion of linkbase information is also relevant to content stored on a DVR or other local storage, and such linkbases may be stored with the video content (embedded or separately), or may be obtained from another source.

Video content streamed over a network (such as the Internet) can be treated in much the same way as from conventional TV distribution sources. Embodiments of linkbases may be embedded, in separate streams, or separately sourced.

TV Programs As Hypermedia Resources

It should be noted that some issues arise with regard to the unique identification of TV programs (and similar non-Internet resources) relating to ambiguity and imprecision in conventional naming and locating methods. These issues relate to the identification of link arcs originating from a TV resource, and also relate to the reverse problem of linking to a specific TV program. Unlike Internet URIs, URNs, and URLs, a reference to a TV program by channel and time may not be precise and unambiguous, since it refers to a time slot that only loosely and unreliably corresponds to a specific content resource entity, which may be broadcast early or late or not at all. Current issues in identifying TV in a hypermedia context are summarized in TV Broadcast URI Schemes Requirements (W3C Note 21 October 1999) which distinguishes a four layer hierarchy of service, event, component, and fragment and two dimensions, one schedule-related and one content-related. That note describes locating methods based on EPG-style channel and time, by query to a service based on a partial description, by reference to stored resources, and various other cases relating to data carousels or encapsulated IP datagrams, as well as others, and notes that "technology-dependent" content identifiers such as SI (system information) data in the broadcast system are not satisfactory. However the problem remains that the vagaries of broadcast may make standard schedule-related identifications unreliable for content identification purposes. Broadly speaking, much of the difficulty comes from the history of broadcast as a channelized push medium oriented to appointment viewing under full distributor control, as opposed to hypermedia as primarily open pull under user control, and it can be expected that the two will be harmonized, with both orientations being supported in an overall context that is more Web-like.

It can be expected that globally unique content identifiers, such as, for example, CRIDs, ISANs, or V-ISANs, will ultimately be usable for TV programs, and that in the interim, useful

sets of non-standard and non-unique content identifiers used in various broadcast systems, such as SI information, can be used in conjunction with system identifiers and other supplementary information to precisely identify specific content resources. For purposes of ITV and similar hypermedia, such precise identification (serving more or less effectively as a universally unique identifier, UUID or globally unique identifier, GUID) may be practical and desirable, even if system specific, and it may be desirable to employ such identifiers for coordinating systems and device sets in accord with the methods described herein. Thus in identifying the state of a TV system, it may be desirable to obtain and transfer such SI or similar information, and in some cases the use of time and channel identifiers may be unnecessary.

This issue of precise resource identification may be important to precise control of coordinated viewing. Simple channel or stream-oriented identification of a resource may not well suited to providing specific control of a resource that is to be presented on a transfer. For example, such identification may not be fully deterministic as to whether the resource that will be found in a channel or stream at the time of activation (or at a specified time) will really be the resource that was named, or some other resource that happened to appear. For example, a request to transfer the same channel while watching one program, might occur as a program change (or commercial interruption) occurred, and cause some other program to appear instead. In simple cases of continuing viewing, this may not be a serious problem, but if the desire was to begin interaction with a program that just ended, the desired function might be impossible. Thus in linking to an enhancement with another device set as target (especially if using third-party arcs), it may be desirable for the behavior to ensure that the current program at the source device set be treated as the starting resource for a link traversal to be completed at the target device set. Similarly, as noted above, and as is a known problem with DVRs, a request to activate a resource

at a given time may not obtain the desired resource. (This is different from the problem of URLs disappearing or having changed content, in that the URL precisely corresponds to a single, entire, discrete resource, even if the identity of the resource is not guaranteed to be invariant. Further, when a resource is substituted at a given URL, that is usually intended to be treated as a valid substitution.) By using hypermedia-style link arc references that identify programs by a precise logical or physical identifier, these schedule-related ambiguities can be avoided, and more desirable and stable linking behaviors may be obtained.

A similar issue relates to the expected behavior on loading a resource. In a TV context, it is generally assumed that one tunes in to a program in progress in real time, and starts viewing from some more or less precise "current" point, except in the case of the newer, secondary alternative of video on demand. In a hypermedia context, it is more common to expect the reverse, that a video resource will be viewed on demand, from the exact start, with realtime streams being a secondary alternative. A harmonized model might provide for consistent default behaviors, using consistent rules for whether a program is activated from the start (such as time=start) or from a current point as default (such as time=now), and for whether a user can override that, in an effect similar to using a DVR to time-shift and pause broadcast programs. Resources might be identified as having either a realtime or on-demand presentation type that is coded in by author/producer/distributors, and which may be subject to override by the user. For example, a live Olympic broadcast might be normally activated in realtime, but with a simple command variation, activated at the start of the program. Similarly, alternative camera angles might default to realtime, but have overrides to start at one or more reference points, for an instant (or not so instant) replay effect. The same can apply to loosely related or unrelated enhancements, some of which may default to realtime, and some not. A simple convention in

link appearance or in link activation controls might be used to indicate to the viewer whether a link was coded for realtime or on-demand entry, so the user could better decide whether to accept or override. Such codings may also indicate whether the realtime start is in reference to some external broadcast or other event, in which case it might revert to non-realtime at the end of that broadcast interval, or in relative time reference to some other stream, in which case that relative reference might remain in effect. Thus tuning to the Olympics the next morning might start it at the beginning of the previous night's program, but any internal relative references might play in synch with the reference (such as alternative camera angles synchronized to the main program). These codings might take forms such as time=now-absolute, time=now-broadcast, time=now-relative to resource name. With a full set of codings and browsing controls, any combination of link traversals and time bases can be provided for (with control by user, author, or some combination), such as, for example 1) from stream A (at time=t), actuate stream B from either time=t or time=0, on either the same or a designated device set, or 2) from stream A (at time=t), actuate another presentation of A on another designated device set, at either time=t or time=0, or 3) from stream A (at time=t) on the TV, actuate linked stream B on the remote control/tablet, at either (specified) time base, or 4) from stream A (at time=t) on the remote control/tablet, actuate linked stream B on the TV, at either (specified) time base, or 5) from discrete resource C on the TV (such as an enhancement text screen), actuate some resource D (continuous with any specified time base, or discrete) on the remote control/tablet, or 6) from discrete resource C on the remote control/tablet (such as an enhancement text screen), actuate some resource D (continuous with any specified time base, or discrete) on the TV, and any other similar combination. Similar controls and codings can recommend, override, and determine how presentations of starting resources are to behave after a traversal, such as whether they are to

continue running on their current time base, or to be paused while enhancements or alternative resources are viewed, and then resumed from the last time position. Such a continuation could give an effect such as that of a conventional advanced TV change of camera angle while such a pause could give an effect such as that of an interpolated enhancement as defined above. The use of time parameters as just described, or using similar methods, enables specification (by user and/or author) of a rich variety of behaviors, including all of the variations after Behrens listed above.

It should be noted that in certain embodiments synchronized enhancements might be based on resources, such as HTML pages, that could ordinarily be discrete, but that have a time-based synchronization imposed on them that could qualify them as continuous resources. With regard to the control methods just described, these can be considered as having elements that behave as discrete resources within the bounded time segments in which they are active. It may be desirable to give the user the ability, using controls similar to those described above, to control whether such resources are presented in accord with their suggested synchrony to a related base program, or to decouple that time-based behavior to support asynchronous browsing.

These methods can be understood as supporting viewing of streams in "hypertime." Hypertime can have flexible connection to realtime, and hypertime for each resource may be related to hypertime for other resources in flexibly linked ways. Just as a user traverses a hypermedia resource tree in (path) ways he defines, based on recommendations and options that are authored in, he traverses hypertime trees corresponding to those resources in (temporal) ways he defines, based on recommendations and options that are authored in. Details of an exemplary set of hypermedia timing controls are provided in the W3C documentation on "The SMIL 2.0 Timing and Synchronization Module." This can be contrasted to conventional advanced digital

TV, which includes multistream elements such as alternate camera angles and synchronized enhancements that can be switched to or swapped, but without true hypermedia browsing controls. In that case the user can only select viewing options as a passive observer, with no temporal control, selecting among views that occur independent of his observation (except for the possible option to apply trick play functions once such a view begins). An author similarly has limited flexibility to offer or recommended options, and essentially edits elements into a single, fixed temporal stream (or set of parallel streams). In hypertime, the user can draw on rich and flexible author-coded recommendations, and, if given suitable browsing controls, can actively determine whether available views are activated and "real-ized," and on what time-base, in a customized, personalized reality. Hypertime is defined by its navigation and control path, and is stateful with regard to time, involving rich, multidimensional state in terms of specific resources, time-bases, time reference linkages, and history and path of resources viewed. Conventional advanced TV simply flows as streams, and is generally stateless with regard to time -- apart from a simple channel history, state information is not material (except to the extent that DVR/VOD functionality provides a simple, limited offset, or "time-shift" from realtime). Hypertime can be understood as essentially a full virtual reality with regard to time (and may include natural video and realtime elements), while conventional video is medium that is essentially in realtime.

Consistent with this, EPG functionality is grafted onto advanced TV as a different kind of element, one that is not a stream (and does not contain streams) but which can point to a stream. In hypermedia, EPGs are just compositions of resources: EPG functionality is just a link from a resource, resources can be streams or discrete, and such links are essentially the same whether they come from streams or from discrete resources like text.

System and Communications Configuration Alternatives

According to embodiments of the invention, the specifics of the communications among coordinated devices may vary depending on the available and installed network technologies. As described, it may be desirable that embodiments be based on full-function home networks, conforming to standards such as UPnP and HAVi or the like, and using systems that include (or are modified to include) software support for the methods described herein. These networks may offer open and potentially universal connectivity, along with rich support services, and such systems can exploit that connectivity to provide the desired MMUI capabilities, as described. Unfortunately, wide use of such network solutions may not be readily available due to technical and business constraints, and alternative configurations may be needed to provide the desired MMUI functions. A variety of representative cases and methods relevant to ITV services are outlined in the section just below, and it may be apparent that similar methods can be applied in other variations.

Consideration of alternative configurations can be viewed in terms of at least two dimensions of variation, one being the hardware, software, and communications configurations of the relevant systems to be used to effect the MMUI at the user location, and another being the nature of the hypermedia or ITV service with regard to how resources are organized and accessed. Details of what state is needed and how it is transferred may vary somewhat from case to case. Further, in some cases, common access to resources by all cooperating systems may not be a standard capability and may require special support or work-arounds.

With regard to base TV and video content resource sourcing alternatives, some possible cases include:

- TV: Sourced through the TV source (such as currently intended for TV use)
- IP: Sourced from the Internet (such as currently intended primarily for PC use)
- Stored: Stored forms on disk, DVD, DVR -- generally follow similar patterns to original sources, but may introduce additional issues
- Flexible (parallel-source): Proposed herein as combined sourcing through either the TV source (broadcast, cable, satellite, etc.) or the Internet, as well as possibly also on stored media, and preferably designed to be used on either TV or PC -- this may include IP streams within a DTV/DVB context.
-

With regard to enhancement content resource sourcing alternatives, some possible cases include:

- TV: Sourced through the TV source, intended for TV use
- IP: Sourced from the Internet, intended for PC use
- Stored: Stored forms on disk, DVD, DVR, could follow similar patterns to original sources, but being stored may introduce additional issues, such as loss of live interaction with others
- Flexible (parallel -source): Proposed herein as combined sourcing through either the TV source or the Internet, as well as possibly also on stored media, and preferably designed to be used on either TV or PC.

The sourcing of the base TV or video and the enhancements may be common or mixed. One mixed case is that of TV sourced video with Internet (IP) sourced enhancements. It should also be understood, that the line between TV and IP sourcing can blur when IP is used within a

TV distribution infrastructure, such as embedded into MPEG streams, or in DOCSIS, or the like. For purposes of applying the methods described herein, key variations relate to whether both kinds of content resources are directly accessible to all systems being coordinated, and if not, whether one is to be relayed to another, or the systems may be limited to presentation of only the resources they have independent access to. For example, current 2-box ETV service offerings represent a case where the video is generally viewed only on the TV system and the enhancements are generally viewed only on the PC system, unless the PC is configured with a TV tuner as a PC-DTV combination system. Various combinations of the methods described herein can be used to add increasing levels of coordination to such offerings. It may be desirable, however, to provide flexible parallel sourcing, by making equivalent enhancement resources available via both the TV signal and the Internet.

With regard to the hardware, software, and communications configurations of the relevant systems at the user location, some representative possible cases include:

- ClosedSTB: A proprietary and closed STB without special provision for obtaining state information and with constraints on modifying software to add MMUI functions (typical of current STBs)
- OpenSTB An open STB (or advanced TV) that offers all STB functions and allows coordination functions to be included with flexible linkage to PCs via gateway capabilities (possibly as a third party modification), such as an STB conforming the OpenCable specification and including its POD module capabilities, or the like.
- CooperatingSTB: A closed STB that cooperating operators and vendors may permit to have the same coordination features as might be provided in the open STB

- LiberatedSTB: A combination of a closed STB with external facilities to extract or duplicate state information to support MMUI coordination, possibly without any cooperation from STB manufacturers or system operators
- ControllableSTB: A closed STB with the addition of signaling terminations for remote control signals to facilitate relaying of control actions, and for non-composited video-out and enhancement-content-out to facilitate relaying and filtering of content. (Such a case might be desired by system operators to facilitate external provision of the coordination function, without risking compromise of access controls or software integrity.)
- PC-DTV-Open: A PC that drives a TV, with OpenSTB function, which can be used as equivalent to OpenSTB, but can also provide full PC function directly integrated with TV functions, allowing a centralized coordination embodiment (PCTV-Central)
- PC-DTV-Closedout: A PC that drives a TV, but which (lacking OpenSTB with POD module features) cannot receive encrypted cable or satellite services (except as slave to a separate STB) (typical of current PC-DTV systems)
- Coordinating-Remote: The usual dedicated remote control could be replaced by a PDA (or tablet or other PC-like device) that has IR (or other suitable) signaling capability and is programmed to act as the remote, thus allowing integration of the liberator functions with the remote control and optionally with the target PC/PDA device (i.e., itself or another device).
- DVR/Gateway-Controlled: In TV configurations with a DVR or full-function gateway, the primary control of video viewing may be from that external device, allowing the coordination function to be placed there, as described above.

Issues in selecting among such embodiments involve a wide range of technology, business, legal and regulatory issues, especially with regard to the tight control currently exercised by system operators over the STBs that connect to their services. Embodiments may be highly dependent on their decisions (and the level of their continuing power) in this complex business/regulatory environment.

It will be apparent that the suggested capabilities of the OpenSTB, CooperatingSTB, and PC-DTV-Open configurations can readily support the addition (primarily in software) of methods of providing MMUI coordination functions described above both for unrestricted broadcast TV, and with full access to closed cable and satellite systems using the conditional access features of the STB or an associated POD module-type device. Such access to cable and satellite service resources may be highly desirable to a commercially attractive ITV service. The DVD/Gateway-Controlled configuration is also readily extended to coordinated use, again primarily in software.

The LiberatedSTB, ControllableSTB, and Coordinating-Remote configurations are representative of new hardware configurations and associated methods that provide alternative ways to obtain a range of more or less similar functions using current or future equipment that constrains access to signals, resources, state information, and controls. These methods generally involve measures to:

- intercept control inputs from the remote control to the STB, and inject emulated remote control signals to the STB
- apply logic to model the state of the STB based on the available signals, in order to drive the STB export/import function

- optionally, obtain program content signal outputs from the STB and relay them to the external device set.

Such external methods can provide the functions that might have ideally been built into the STB. The objective is that they do not impede the function of the STB, but add intelligent communication and coordination functions that the STB does not provide for itself. One class of these methods, represented by the LiberatedSTB, is based on use of an external device that intercepts signaling from the remote to the STB, such as by using IR sensors and lamps placed near the STB, much like for current VCR commander cables from STBs or Cable commander cables from VCRs. Serial cables may also be usable as a control input to the STB, as can HAN connections. IR input directly to the STB could be shut off (such as by physical masking) to avoid uncontrolled signals. The liberator device could have sufficient intelligence to model the state of the STB, at least with regard to the current channel being viewed, and possibly also other control details as well, and to communicate that state to the coordinated device set in either push or pull mode. That forwarding might be by IR, wireless, or other suitable means. Similarly, the liberator device could be responsive to interpret requests from the coordinated device set to issue IR, or serial, or other control commands to the TV/STB, such as to change channels or control trick play functions or other advanced features, as appropriate. It may be expected that over time, remote controls may shift from IR to Bluetooth or other wireless HAN or PAN signaling, and it should be understood that all of the configurations described as relating to IR are meant to be adaptable to such signaling, and as will be apparent to one skilled in the art, they may be somewhat simplified as a result of the greater flexibility and ubiquity of such network connectivity.

The ControllableSTB is suggested as a new class of STBs that make varying degrees of limited provision for coordination function external to the STB. This could permit STB vendors and system operators to enable coordination to occur external to their systems, with well-defined inputs and outputs and strictly limited exposure. Such a configuration is also especially advantageous for the case that enhancements are embedded in a TV transmission, and thus not readily accessible directly by another device set (such as through lack of a tuner and/or lack of conditional access rights). Outputs from such a device could include 1) detailed control event streams to allow external interpretation, modeling, and reaction, 2) TV signal enhancement trigger or linkbase data, and 3) full relay of embedded enhancement streams. A method for relay of all such data could be to convert such data to an IP format and forward it via LAN (wired or wireless), but other output signals and connector types may be used, including basic A/V out signaling and use of the 1394 Fire-wire connector included on advanced STBs.

One simple form of relay is to provide a slight modification of a standard STB to open it minimally by providing intermediate outputs for use by a liberator device. The liberator could be a separate device, or integrated as a hardware or software-only adapter into a PC or other system to be coordinated. Digital TV set-top boxes are routinely equipped with facilities to connect to a TV source (typically analog or multiplexed digital QAM) and convert (tune, demux, decrypt, decode, and compose to RF output) that signal for use with a TV monitor, but do not have facilities to make that signal available in a form suited to further processing by a PC. The STB uses an internal CPU to composite additional elements onto the base video, in response to user interactivity. The box outputs the resulting composite A/V, but not the base program A/V or the raw supplementary data obtained from the broadcast signal. It may be desirable to enable selective pass-through of those additional elements to shield the set-top box from the interactions

on the PC/PDA system, and separately apply them for the PC/PDA, so that there is no change to the base program A/V on the TV (and further, to allow separate interactions at the TV). Preferably the PC/PDA interactions could generally not be passed back to the set-top box (except when a transfer to the TV is intended), but rather processed at the liberator, which could then use a separate path to get individualized ITV enhancement responses without interference to the main TV screen. It may also be desirable that the liberator could separately obtain both the basic video and the additional elements (in raw form) from the set-top box, and act as an intelligent session-sharing device for the TV and PC/PDA. (Separate interactions intended for the TV could be handled normally by the STB.)

Thus various combinations of ITV outputs for a given program that could be useful in different embodiments are 1) the standard A/V output with the base program only, for basic TV, 2) a demuxed and decrypted MPEG or similar stream that could give an external liberator or other device access to the content needed for full independent interaction (which might be output to a second A/V connector, or via LAN), and optionally, 3) a composited signal with the base program, ITV bugs, plus selected enhancements directed to the TV-out as an alternative output, for use when interaction is intended to target the TV (which might be output at the standard output connector). This process could be done separately for each of 1) the basic TV signal and 2) the supplementary digital interactive control content that might be coded in a VBI or DTV format, and 3) any of multiple PIP elements. Such separation can enable the user to select various combinations of output to either TV monitor or PC (via the liberator or adapter system). Thus with full pass-through, either device could enter inputs and both devices could show a common result screen. With selective pass-through, one useful setting could be for PC

interactions to cause change only to the PC display, so that it could be used for interactive content supplementary to the TV program without interfering with the display on the TV.

Similarly, a liberator box could relay remote control signals to the coordinated device set using the same kind of IR or wireless signaling as the TV remote, as has been suggested, but it may be preferable to convert those signals into an IP format for communication with the PC over the LAN (802.11, Bluetooth, etc.). This could eliminate the need for the coordinated device set to receive IR, and can also eliminate the need for such a device to send it. This is useful, whether for relay, or simply for a PDA or tablet to act as a remote control without the limitations (distance, line-of-sight, reliability) of IR signaling. Note that the use of IR signaling for remote controls may desirably be replaced by such wireless RF facilities as ultimately being more powerful, economical, and flexible, just for its conventional functionality, and this could further expand the range of devices that are readily coordinated.

A configuration suggested as the Coordinating-Remote type can be applied as a straightforward variation on any of the prior cases, one in which the function of the remote is subsumed by a PDA (or tablet or the like). Here the signaling control need not be inserted between the STB and remote control, but can be integrated with the remote control device. The task of making the PDA work as a remote follows the existing example of the OmniRemote software for the Palm. Relaying to the PDA (if the PDA is the target device), and shielding of the STB from undesired interaction signals can all be done internal to the PDA. In simple such embodiments, the only need for special support at the STB is for cases where a content signal must be passed from the STB (the TV-sourced cases), and such a device could be simpler than the ones that handle control signaling as well. If the target device to be coordinated is some

system other than the PDA that acts as the remote control, then relay functions to that device from the PDA/remote could be included as for the above cases.

As noted above, the DVR/GatewayControlled configuration can be useful in situations such as where a DVR acts as the primary control for a TV system. In this case the DVR acts much like a liberator in handling the remote control and driving the STB, and is thus fully aware of and able to control the key aspects of state in a straightforward manner. The DVR can be connected to the PC/PDA device through any convenient method. Note that DVRs may in some cases be integrated with the STB, and in some such cases they may be more closed in regard to extensibility.

An additional variation of interest involves the use of voice control, based on speech recognition, as a replacement for a conventional key-driven remote control. This can be done with portable devices such as the KASHNGOLD InVoca Deluxe, but may also be based on other devices, such as a PC using software such as AUTOMATED LIVING HAL2000. Voice activated control based on a PC could be used to drive a TV/STB, and could optionally also drive functions on the PC. Thus such a PC can provide the signaling and state management functions for the TV/STB much as described in connection with the liberator, using only a control connection to the TV/STB, such as IR, serial, or wireless LAN. All of this could be placed in a laptop or tablet configuration, but for reasons of size, packaging, physical connections, power, and the like, it may be desirable to split functions between such a portable device and a base unit, which could be another PC or a special device. Such a split could be achieved through peer-to-peer connections (preferably wireless), or using a client/server or I/O terminal structure, such as ones like those discussed above in connection with Mira. Such a PC-based coordination method could be used with a conventional remote as well, by linking the remote to the PC instead of the

TV/STB. That could be somewhat awkward with regard to the limitations of IR signaling, but could be more suitable for a remote using other wireless networking.

Given the range of embodiments addressed here, some review of key aspects of the basic methods may be helpful. Provided that content and control signals are obtainable as described for each case, the common task remaining is the higher-level control logic that implements the exporter/importer for the STB. (The import/export at the other, computer-type device set is largely independent of these variations.) This requires modeling of the state of the STB, as a surrogate for direct access to its internal state. A state model is maintained by keeping track of all control actions, and applying them to a state machine model that simulates relevant aspects of the logic used by the STB to present the basic TV and enhancement content. Such models can be created for each STB type and each ITV programming/sourcing format. This logic can be placed within the liberator device, or can reside remotely at the target system. The latter may be preferable in terms of hardware cost, and availability of software and network resources to facilitate programming and support. In that case, the liberator acts as a simple peripheral device that relays control signals under the control of coordination software in the target system that manages the control of both device sets.

Granularity issues also apply much as described previously. Coarse-grained embodiments that only permit migrations at a few well-defined interaction entry/exit points are simplest to accomplish, and may be best suited to the constraints of a liberator embodiment. The mixed access case of TV-sourced base video plus IP-sourced enhancement resources could reasonably involve limited support for interaction at the TV/STB, so the granularity might be correspondingly limited. The simplest such cases could just pass the current channel and

optionally an activation signal when interactive functions are desired, and could also optionally add relay of TV-sourced trigger events.

It should also be noted that coordination with regard to EPG content (which might be carried in the VBI or DTV broadcast or sourced separately) is particularly simple because of the well defined and limited functions required and its limited, fixed points of coupling to the TV content, so shifting those functions to an alternative device set is simpler and might be done apart from the more general coordination functions.

An alternate embodiment of the communications required for basic coordination that may be desirable in the near-term is to do this through external WAN communications as noted earlier. This may be particularly relevant to early acceptance of ITV to finesse the problems and delays of funding and completing installation of more advanced systems. Many STBs currently installed are not connected via a LAN to alternate device set candidates such as the PC, but these systems can, instead, be linked via a remote location such as a head-end. These STBs may already have software capable of transmitting current channel and other key viewing event state information to the head-end, such as to enable tracking of viewing, use of services such as VOD and PPV, and targeting of advertising, and are addressable to identify the specific STB device. In that case, no new deployment may be needed, and all linkage and processing needed to provide state record assembly and transfer could be accomplished at head-end server locations. Should that software not be in place, it should be relatively simple to add it as a software-only upgrade to the existing STB devices. Such externally mediated transfers might be most easily accomplished as pull transfers, but addition of simple software functions at the STB (again, possibly using existing, installed hardware) to support activation of push transfers based on simple commands should not be difficult either. At a more fine grained level, specific link

activation and arc data (including ATVEF trigger data) could be sent from the STB to the head-end for relay to an alternate device set (or caused to be sent directly from the head-end).

Such externally mediated coordination could be enhanced by value-added server functions. For example, a cable or satellite operator (or other party) could operate an Internet portal that mediated the communications described by allowing user PCs to log in to a secured account to link to the state information on the TV (for any of that user's STBs) and request pull transfers (or receive push transfers). Such a portal could have a scope that covered all channels and programs obtained through that operator's service, and could optionally include similar data on other resources, such as broadcast channels, Internet video sources, and digital movies (broadcast or DVD, or whatever), and thus serve as a primary entertainment portal for the user. Because of the sensitivity of such detailed viewing data, privacy and security of such data may be vital, and could be assured by both policy and technical measures of kinds well known for such data and Web-based services. The basic functions of such security measures are to identify and authenticate any PC user seeking access to data for a TV, and to use an access control list or similar specification of privileges to determine that the user at the TV and at the PC correspond to one another, at either an individual or family level, as may be desired, or are otherwise to be granted access privileges, with support for the case that different ID schemes and authentication methods may be used at the different device sets. Since basic Web HTTP services are oriented to pull requests, support of push transfers from the TV and of advanced tracking and synchronization could be accommodated with more advanced techniques using HTTP or alternative protocols, such as are commonly used in Internet chat and collaboration applications, or like those used for ATVEF trigger reception. Such chat and collaboration protocols include, for example Internet Relay Chat (IRC), Short Message Service (SMS), Simple Mail Transport

Protocol (SMTP) Jabber, Wireless Village, and proprietary instant messaging networks such as Yahoo!, Microsoft Network (MSN), ICQ, and AOL Instant Messenger or NetMeeting and T.120. Such use of a portal to link a TV to a PC can also provide other benefits by making resources of the PC usable in conjunction with enhanced TV services, such as to draw on information from cookies and other PC-resident-data and applications (subject to suitable security measures), as described further below. This can reduce the need for information to be obtained from a user at the TV, and exploit the ability of the user to easily maintain personal profile and support data at the head-end/portal from a PC. In cases of multiple STBs in a household, such as in multiple rooms, or of advanced STBs that support multiple TV sets, including STBs with gateway functions, such relay associations could be specific to any selected TV set, and similarly, multiple such relay processes could be concurrently supported with any of multiple PCs.

Such relay via the head-end can involve transfers in either direction. In addition to state transfers from the STB to the PC, user commands at the PC could drive actions at the STB, again with simple software enhancements that can be provided at conventional STBs and head-end servers. Aside from the advanced case of general ITV or hypermedia session transfer, simpler transfers could permit the PC (or any other Internet-connected device) to serve essentially as a remote control to the STB for specific functions, such as, for example, channel changes, camera angle or other viewing option changes, PPV or VOD requests, and provision of EPG functions at the PC that control operation of the STB. This method of relay through the wide area network is also applicable to the more general relay of TV-related signals (streams) from a STB to other intelligent devices that is described below (for transmission via IP over local facilities).

Another embodiment of a relay facility is suited to a likely configuration for near-term connectivity that may precede the availability of full-function home networks. It may be

expected that STBs may, in many cases, integrate with cable modems (using DOCSIS or similar protocols). Such combinations can also be expected to offer wireless IP LAN connections, using technologies such as 802.11 (a, b, g, or other variations) or others such as Bluetooth (or similar PANs). Such facilities may lack the rich HAN and gateway functions promised by UPnP and HAVi, but could enable direct local communication of STBs to PCs and other similar devices using facilities that can be expected to be in place to meet the conventional individual Internet access needs of such systems. Because the cable modem may in some cases be integrated with the STB, it can be expected that they may be designed such that IP connectivity between the two will be available (or readily added with software only). In that case, such a configuration could provide the necessary path for local coordination, allowing transfers to be pushed or pulled from the TV to the PC using this IP path.

The above discussion gives many examples of how the coordination services described herein can be adapted to a range of hardware and network variations that may be suited to current and future systems. It will be apparent that other similar adaptations can be made using combinations of these or similar methods and configurations to adapt to meet other needs and work with other configuration alternatives.

According to embodiments of the invention, the signal relay function addressed in numerous variations above is also suggestive of a more broadly useful method for extending a STB or similar TV receiver device to convert the TV signal to a digital stream, such as in MPEG format and carried using IP, that can be passed on to another system through widely available home network communications. This could be in the set-top box or a separate liberator, or in a home gateway, and could be linked by any LAN or wireless technology, such as 802.11 (Wi-Fi) or Bluetooth. For STBs lacking wireless support, the 1394 connection could be used to attach an

external wireless transceiver device, and this could be a desirable way to enhance installed base STB hardware. This content may include the primary video signal, or any secondary video signal selected during an interactive session, whether an alternative live feed, or a stored video or multimedia program. This relay could be usable in support of coordinated viewing and interaction, and supplementary to the coordination of remote control interactions outlined previously, using currently available PC/PDA devices. It could also be usable in support of uncoordinated use of a PC or tablet or any other suitable appliance (including additional TVs) to access ITV programming, without the need for special hardware or direct connection to a set-top box or other broadcast TV source. Such a single point of conversion could reduce expense in the home by eliminating the need for TV receiver hardware in alternative device sets, such as a PC/PDA. It could also reduce wide area network bandwidth requirements for access to broadcast programs by avoiding the need for video streaming over the Internet (or private IP unicast or multicast facilities) when an existing airwave, cable or satellite broadcast channel is already carrying the program. Such relay is essentially a home gateway function and could be part of the STB, or inserted between the set-top and the TV monitor, or elsewhere, and could be transparent to existing devices. It could obtain the TV signal, convert it to an appropriate streaming format such as IP, and output it over the home network to any device that wants it. Thus it serves as a central/shared receiver of TV, from a source that may be non-IP, that can convert that program to an IP format for use by other devices in the home. This uses the TV transmission plant for wide-area communications, uses a single TV receiver facility in the home, and efficiently distributes the program in IP format within the home for use by any IP-capable device. For example, using such a relay to insert a broadcast video window into a conventional Web page could be more efficient than using IP streaming to obtain the same video over

conventional Internet paths (and could do so without the need for a tuner in the PC). As will be apparent to one skilled in the art, various decoding and decompression facilities can also be provided by this common receiver device, to transcode to a different compression scheme, or no compression, for local retransmission in the home, and various DRM and conditional access features can be applied to limit uses in accord with access rights.

Sensory Mode Transfers and Heterologous Modes

Most of this disclosure focuses on cases of transfers that change the locus of work and the use of different device sets having homologous modes, but the case of changes of heterologous sensory mode may also be important and the same methods can be adapted to such cases, as will be apparent to one skilled in the art, based on the teachings herein.

For example, such a transfer might occur between a speech-based browsing segment using a phone, and a display and button-based browsing segment using a PDA or PC. A user might call by phone to check on his scheduled flight departure, find that he will not make it, and verbally request that the "reschedule my flight" link be traversed not to the phone, but to the PDA, to allow for easier scanning and booking of alternatives. Instead of re-identifying himself and the flight he wants to change, the browser could be opened to the reschedule page with that context re-established. Conversely, the user may confirm a flight on the PC, and traverse the confirmation link with the indication that the confirmation response is to be targeted to his phone, so he can run off. It is noted that the methods described herein do not require special coding of fixed alternatives at fixed points in specific resources, such as by placing explicit alternate buttons on the confirm-this-flight checkout page for: confirm to Web, confirm to e-mail, confirm to Web-phone, and the like. The methods described here instead make such

options standard in form routinely available at any traversal point in a session, with full plug-and-play access to all available and suitable device groups.

It may be helpful to clarify some issues relating to emerging requirements for synchronization across multiple heterologous modes, particularly cases that explicitly combine speech with other modes, such as for mobile use. This objective has some parallels with the objectives that have been addressed herein, but might differ in that primary concerns might relate to tightly coupled, fine-grained, synchronized coordination of multiple simultaneously active heterologous modalities of input and output, and to the special issues of synchronizing GUI interactions with speech input and outputs, based on such complex mediation processes as speech recognition and text-to-speech. This might involve complex synchronization of events and browser activity and use of advanced methods to provide conceptual translation and alignment of related resources (such as speech segments and Web page text) between these very different modes. To support this, it might be desirable to develop new markup for hypermedia resources that makes explicit provisions for simultaneous use of multiple heterologous modes, creating a tightly coupled, unitary experience. Such a unitary experience might be expected to be oriented to support of a single user task, and at any given time might typically involve a single common resource, or a set of equivalent, parallel resources, simultaneously presented in multiple heterologous modes, that is correspondingly responsive to inputs from any of those modes. Similarly, ongoing navigation might be ganged so that any mode of input applies to all active heterologous modes in parallel. Thus in MVC terms (as described below), the tightly coupled heterologous modes might present multiple views of a single common or replicated core state model, linked by tightly coupled or fully integrated view controllers. In contrast, homologous device sets differ to some moderate degree in adaptation, relating to issues such as display

resolution and fineness and richness of input controls, but are inherently similar in operation and in their presentation of hypermedia resources. The markup methods addressed herein are directed primarily to this case where alternative homologous device sets might be used in sequential sub-sessions or with limited simultaneity, creating a loosely coupled experience supporting related but disjoint activities or tasks. Such disjoint, loosely coupled experiences are supportive of multitasking (or task transitions) by the user, and at any given time might typically involve different (and non-equivalent) resources at the different device sets, which might have varying levels of correlation and/or linkage of activity depending on the user's task, navigation path, and work style, and in which inputs relating to a given task and/or sub-session are typically restricted to the device set presenting the resource that addresses that particular task (apart from commands to transfer a session activity). Similarly, ongoing navigation might desirably have an affinity to one given device set unless stated otherwise. Thus in MVC terms, the loosely coupled homologous modes might present largely independent views of disjoint, but typically more or less correlated, state models, using what might be logically separate view-controllers. Nevertheless, the UI methods taught herein and the UI methods that might be developed to support that kind of tightly coupled and heterologous multimodal sessions could be adapted to align with one another, and, as will be apparent to one skilled in the art, many of the methods taught herein could also be applicable to those objectives as well.

According to embodiments of the invention, methods relating to heterologous multimodal browsing might further involve use of a browser architecture that could be based on a single, central MVC model of interactions that could be used by multiple browsing view-controllers,

and which would thus be unlike common browsers. Such a method might employ techniques used in some fully synchronized collaboration systems.

Alternately, the methods that have been described herein might be employed, for example, as add-ons to conventional browser architectures. Such embodiments might not require a single central mode, although one could be employed if desired. It is further noted that the event stream simplification strategies described below could also be beneficial to such heterologous multimodal browsing.

The methods described here are not specifically oriented to context-aware applications, but it will be apparent to one skilled in the art that they can be useful in such embodiments, and can be extended accordingly. The methods described herein have been described primarily with regard to aspects of context (state) that are internal to one or more systems available to a user, with some limited input and inference regarding the user's intended device set usage and task characteristics, and how to exploit those systems and that state information to provide coordinated application functions. Naturally, to the extent that more extensive external context information (such as from sensors and physical objects) and corresponding application support functions are available, these methods will generally be applicable to those state data and functions as well. For example, sensor data on the movement of a user might be used as a cue to transfer a session from one device set to another, and more specifically, this might include cases such as pausing a session if a user gets up from his seat, or transferring/activating an enhancement session if a user reaches for a tablet.

Model-View-Controller Embodiments and Event Stream Simplification Methods

The methods just described apply equally to conventional browser designs, and to possible new designs based on a model-view-controller (MVC) architecture that separates functions in terms of an underlying model, which encapsulates application state and includes the hypermedia storage layer, a view that presents that to the user and obtains inputs, and a controller that defines behavior and responds to user interactions with the view to cause changes to the model. Such architectures may simplify support for features such as device independence and collaboration, as well as multimodal features, by isolating such issues from the model.

The variations described earlier relating to multiple systems and independence of systems and browser instances apply to MVC architectures as well. MVC designs may provide for coordination of complex and distributed multi-device-set and multisystem browsing based on coordination or synchronization of separate model instances, in which each instance contains state information for the device sets and processes it supports, or alternatively by using a centralized model that contains all state information, as the driver for all views and controllers. In the case of the centralized model, as for the single browser instance, coordination is relatively simple. This can also be thought of in terms of a single centrally controlled application program.

For the case of distributed models, just as for multiple browsers and independent systems, the coordination requires more attention. Reviewing the methods described herein in terms of MVC concepts is useful, both to clarify how the methods apply to an MVC architecture and to further clarify the conventional case. Two alternatives are that the exporter/importer/tracker be built as 1) an adjunct to the model that performs exports and imports in support of transfers to other models (essentially the case already described), or 2) alternatively as a mediator that intercepts (or mediates) controller actions as state change events, and broadcasts them to other

coordinated models. The first can be done essentially without any preparation prior to the transfer request, as an ad hoc collection and export of current state data from the browser (the model). The second involves ongoing collection of the event stream that drives state changes as they occur (which can be more or less independent of the internal structure of the browser/model), but which may be limited in its ability to transfer all state information needed to perform sufficient input re-feed to correctly initiate a synchronized replica, to the extent that the collection process is not started prior to relevant state-change events (at least those that are not fully reflected in the hypermedia storage layer). Thus the second may be architecturally cleaner, but may require enabling tracking well before any transfer, which can be problematic in terms of performance and usability. Both of those alternatives can be applied to either MVC or conventional architectures. Both involve an export/import activity, the first being essentially a consolidated batch export/import of all relevant state at once, the second being an event-by-event approach. The methods described above allow the burden of full logging of all events to be reduced by limiting state export/import to the aspects and times needed for transfer of work locus when requested. Thus all models need not be synchronized at all times (which is unlike the case for collaboration or for fully simultaneous use of multiple modes). Further detail on these state tracking simplification and reduction methods is provided below.

Features and Functions

As noted, these methods may be applied with a variety of triggering conditions, hypermedia systems, user system architectures, and form factors. Details of user interface, link arc coding, presentation and implementation may vary accordingly. Some of the features and functions include:

- Communications support and device set management -- types, discovery, grouping, and standards
- Coordination control center
- Alternative transfer activation/deactivation control methods and standing sessions
- Asynchronous transfers, retention of transient links, and simplified transfers
- User controlled targeting during browsing
- Link attribute controlled targeting for authoring/production
- Priorities and preferences for managing device set use
- Source and destination session behaviors and synchronized tracking options
- Congruent behavior of independent, dependent, and centralized systems
- Transactions, editing, cookies, and peer/server state
- State tracking simplifications
- Security, privacy, and digital rights management
- Hypermedia edit applications
- Venue/kiosk-based coordination
- Flexible and deep integration with other applications
- Sensory mode transfers
- Virtual and augmented reality
- Revenue models
- Comprehensive framework for MMUI browsing

The following discussion explores some representative embodiments of these methods under various usage scenarios.

Communications Support and Device Set Management -- Types, Discovery, Grouping, and Standards

These functions and scenarios may naturally depend greatly on the nature of the systems and device sets available to the user. For convenience, classes of devices with similar form factors and technologies may be grouped as device set "families." Some representative cases include a standard across-the-room TV with a digital STB with a remote, which may be used in conjunction with a desktop PC located within easy reach. Supplementary device sets include a notebook PC or a tablet PC with a wireless LAN connection that can be used from a sofa, and kept handy on an end table. Current notebook or tablet form factors are fairly suitable for such use, and improved designs can be made more convenient, and styled for various home décor tastes. Such a high-resolution device set enables power browsing for intensive tasks. Another device set is a PDA-style form factor that is more compact, easier to handle in a casual setting, and less expensive, but also less suited to power browsing and intensive work. Such devices might be have a charging pad base unit designed for convenient nearby storage.

According to embodiments of the invention, a user may want to have an array of such options available. A major advantage of the methods described here is that they can be embodied in simple software that can be added to any independent browsing-capable system, including legacy devices, to enable coordinated use on demand. The use of modular design and standardized base level communication provides the ability to make it easy to support coordination on an open basis for any browsing-capable system, by simply adding the necessary exporter/importer and device set management software. This eliminates the need for all cooperating devices to be provisioned in common. Any devices at hand can be discovered and

activated, including all systems owned by a user, as well as systems that may be temporarily available, such as PDAs, notebooks, or tablets carried by visitors to a home or other venue. Such devices need not be dedicated to a given coordinated browsing configuration and to browsing use, but can be brought into such use at any time on an ad hoc basis. Thus hardware expense becomes a minor factor, as users obtain various intelligent devices of various form factors for other purposes. Most households can soon be expected to have a rich complement of computers including desktop, tablet, PDA, and cell phone form factors, and any of these can readily be made capable of coordinated browsing use. As coordinated browsing becomes common, families may wish to acquire special living room or den-oriented tablets that might stay on the end table for more or less dedicated use, but that is not necessary. A bedroom system may be used more casually, so a smaller PDA-like device may be preferred, as might be the case in a car, as well. New designs might provide for furniture with build in devices, such as seating with tablets that pull out of an arm, like an airline bulkhead seat or lecture hall fold-away tray, or end tables with hide-away or swing-out devices. Other form factors may also appear, such as special glasses with heads-up display.

In considering device set groups for an ITV system, for example, one group may be the ITV system alone, a second group might be the ITV system with a specific portable notebook PC, and one might be the ITV system with a specific PDA. "Generic" groups could be definable as consisting of families of devices of a given form factor, with specific device sets assigned to corresponding generic groups. This can simplify control and adaptation, such as with regard to preferences and default behaviors, allowing a level of generic specification that is common to all device sets of a given class, with a further level that can be specific to a device set. Thus any similar tablet, for example could be handled on a similar basis. Groups may be defined as

overlapping and having common members, so for example an ITV-centric group could define the ITV system as a primary device set and a PC laptop as a secondary device set, but the same devices could also be members of a PC-centric group that defines the PC as primary and the TV as secondary. The same PC might also be primary or secondary in one or more other groups that might have a PDA, and be used without an-across-room TV. Similarly, device sets may be grouped into subset or variant cases that facilitate adaptation to changes in the presence or absence of specific devices within a device set. This can be useful for example when a tablet may be used with or without a keyboard. Such a structure provides a rich basis for setting preferred and default behaviors that can be invoked on a task-appropriate basis at different times, and that can serve as a context for targeting actions by the user and for coded targeting controls set by authors, producers, or distributors of hypermedia resources.

Coordination Control Center

Some embodiments might also provide for a control center application to package and organize support of coordination functions. This might include access to the various setup facilities, such as for managing device sets and groups, and for generic viewing control and support functions. Such services may be implemented for each participating system. Such services could be provided on a coordinated peer basis, so that no system was dependent on any other single system for these functions, but alternative embodiments could be based on a central master system for a defined scope of systems and services. Coordination of these services could be through any suitable protocol, including the use of the central master system and database, or any peer protocol, and can be based on distributed storage local to each device, or some use of shared storage at a local or remote server, including use of SANs or NAS.

According to embodiments of the invention, viewing control and support functions might include services as an entertainment portal, EPG, DVR-style library or archive manager, and the like, whether provided locally or remotely, and whether integrated into one package or service, or composed of various linked services, such as might be composed with Web services. Implementation on a PC might be as a local application (preferably cooperative with the browser), or as a remote ASP service, possibly enhanced with a browser accessory that provides selected local functions. This can provide a control panel offering functions such as to facilitate access to portal features and to support a pull transfer from a TV or other active device. Such a facility could also provide browsing controls that integrate with the browser for MMUI use.

Alternative Transfer Activation/Deactivation Control Methods and Standing Sessions

Also desirable may be the ability to control targeting by simply activating or deactivating a device. If so indicated in the device set preferences, one way to transfer a session in some cases, such as from a TV to a designated enhancement device set, such as the tablet at the sofa, could be to simply power on the device. The device could join the network, check its coordination attributes (which may be self-contained or obtained from a network source), determine that it is to seek transfer of enhancements to the current TV program (or initiate a new enhancement session associated with the simple video session) on startup, and initiate a transfer pull request accordingly. Thus a user could view the current base enhancements (such as a menu of current selections) for the current TV program simply by pulling out the tablet. In some embodiments, such an activation might transfer an enhancement session already in progress on the TV, transferring its more rich state, and with preset options for the disposition of the

transferred session on the TV. Similar simple controls can also be offered for use of a PDA in coordination with a PC, or for other combinations.

According to embodiments of the invention, corresponding options could also be set on deactivation of a device, such as to automatically transfer a session before termination. For example, it might be desirable to return an active enhancement session to the TV on deactivating a tablet device. Further options might push state to another device set, or to some intermediate caching storage or session proxy system or repository, in a ready but inactive mode, allowing the other device to activate the session at some future time, even if the originating system is unavailable, in what can be considered a pull from cache. By using a session cache proxy in this manner, a session can effectively be swapped out and held in limbo, with no active device sets, until a user requests that it be pulled and activated at any capable and authorized device set. Such a feature can also be useful as a standard browser feature for saving and restarting sessions even for a single device set, whether using an external cache, or one internal to that system, to provide functions far richer than current bookmark capabilities.

A related feature might also be desirable is a similar push transfer (via caching proxy) capability to trigger activation of a target system from the originating system, if the target system is not currently active. This might be desirable to simplify transfer to a device not yet ready, especially in cases such as for a time-specific link that might not be directly actionable by the time the new device was ready. Such a push could be completed when the intended target system was activated and became accessible.

An additional feature is that of a standing coordinated session, such as for a tablet that is routinely available for use with a TV. Settings could provide that the default for this device is for it to maintain a simple enhancement session with whatever TV program is active, and

automatically present the main menu or home page associated with that programming (possibly changing with time-position) to serve as an always-ready interaction device. For such a case coordination could be maintained to allow this standing session to maintain awareness of selected state changes on the TV, notably when a channel change occurs. To support that, it may be desirable to provide a limited event tracking mode (based on a bifurcation of base video versus enhancement activity), in which only channel/program change events (and optionally pause and other VCR-like trick play actions) are transferred. Similar standing sessions could be provided, separately, or in a presentation additional to the program related session just noted, to present generic information from a portal service. Such a second-screen service may be more attractive to viewers than the method proposed by some ITV services of "force-tuning" the TV to a portal screen at power-on.

Note also that the related concept of standing enhancements that are always available from a default source, which source may be defined by a well-known convention (whether or not a standing session is activated), represents an advance from current models in which enhancements are not assumed to be available unless a specific indication is given. As ITV develops, an expectation of enhancement may be the rule, not the exception, even if the default enhancement is quite limited. This goes beyond the idea of always-on enhancements, and the use of virtual channels to create generic portal or walled garden services, to include dynamic, program-specific resources . It may give the effect of a special virtual channel or portal that may be dynamically associated with each of any or all programs.

Asynchronous Transfers, Retention Of Transient Links, and Simplified Transfers

Building on the above methods, the ability to defer a transfer to a future time will now be discussed. Such functionality could allow a user immersed in a current program to initiate transfer actions that could be held in a pending state, so the user could continue viewing the current program without further interruption, and attend to the transferred activity at some more convenient time in the future. Such features may be particularly attractive to program providers who wish to avoid loss of their audience to tangential activity. It will be apparent that this can be provided using the same methods and with a variety of user interface options. Deferral of any number of transfers can be accommodated by saving the transfer state records. Enabling of such deferral could be a standard feature, wherein transfers are routinely held until actuated at the receiving end, or could be indicated by command at the time of transfer. Activation of a continuation session could be automatic, as described above for power-on, by a simple explicit action to enable the next session transfer, or by selection among a list of pending transfers that may be identified by source, time, link descriptor, or any of various other identification cues that could be stored with the state record. It should also be recognized that such deferral of a session might be a desirable new feature even within a single device set, where the transfer is just over time, not across device sets, and that this is readily provided using the same methods.

A related useful feature is the ability to save and permit delayed activation of time-position-specific links, such as for example ATVEF triggers. In current ITV systems, such links become unusable and are typically discarded once their time-scope has ended. Such links could instead by saved in a special history linkbase. A variety of user interface methods can be applied to using such saved links. One is simply to record the entire program in association with the links, to allow replay and delayed activation in the intended context at the appropriate relative time positions. Another is to provide a reduced variation of the program on recording, such a

keeping only snippets of context video, or using reduced resolution video or still reference frames to provide a context the user can relate to apart from the full base program. Another is to just list the links with the context parameters, and optionally with a text descriptor which may be provided with such a link (in a manner similar to the use of an "alt=" text label that is coded in an HTML image-related IMG SRC link for use as an alternative to the image itself in case the image is not presented). To the extent that such links have a time-scope that is coded with respect to real time, they could be adapted to define their scope in time relative to a program start time or current time code or frame number, either directly or through a separate time base offset factor. Numerous variations on these options will become apparent to one skilled in the art. Such facilities can provide an effect similar to asynchronous or post-broadcast enhancements, but with added flexibility to both author and user in determining when or if such links should be available.

While the orientation of much of this discussion has been to transfer of full sessions with relatively rich state, it should be noted that it might also be desirable to provide simpler transfer functions based on simplified variations on these methods. This might be of particular value, for example, for activation of simple transaction activities, where the communication process involved in the session transfer is used to provide a more limited communication to an alternative device set, or just to a remote server. This can be thought of as similar to the case of a transfer at the start of an ITV enhancement interaction, where state is minimal, and little more than the link arc-related data need be transferred. For example, a TV program or advertisement may contain a link to enhancement information or to initiate a transaction, and the session transfer process may be simplified to simply pass on the link arc data. The receiving device set might simply provide a list of received resource links, much like a special bookmark or hot list.

As noted just above, this feature might also be used at the same device set, simply as an action deferral method. In some cases, the transfer action might relate to no more than a simple user request for action by an advertiser or vendor, in which case the methods described here can be adapted to send the transfer state record to a remote server, and the state record is used as a form of transaction message, with no further action by the user required, or with simple confirmation actions only. In such a case, the transfer record might include rich user information, such as billing address, shipping address, and other personal profile data. Such a simplified transfer record can be transmitted using any appropriate protocol, including HTTP, SMTP, SMS, other message protocols, or the like. One very simple variation on this is to trigger an e-mail to the user, and such an e-mail could contain a link relating to further information or actions to be taken. Another very simple variation could generate a Web page containing relevant information, links, and possibly forms for additional input. These relay processes could include application processing that adds transaction-specific elements, either at the STB or at a head-end server, but can also be limited to a standard process that simply packages link arc data and sends it to the PC or any designated alternate device set (with only a basic relay involvement at the STB and/or head-end). This pure relay process could avoid the need for any application or link-specific support at the STB or head-end, and could allow an interaction at a TV to produce an effect similar to that of having clicked a link at the PC (that produces a corresponding Web page or e-mail message). Such methods can also benefit from the user identification and profile data that might be available from the alternate device set, such as in a cookie, database, or application -- data that may not be readily available at either a STB or the head-end (at least not without burdensome user entry). Thus for example, a user could order an information package or product, and have fulfillment and payment details provided by the portal and/or the user's PC,

with any maintenance of user data at the portal managed from the user PC. Depending on the communication configuration alternatives, as discussed above, such communications can use any suitable combination of local and remote paths. Again, such simplified coordination might either be synchronized with the primary browsing session, or asynchronous. It should be noted that this kind of simplified transfer can be useful with current low-end STBs, in communication with a head-end server, as described earlier, and that, depending on the particular type of request to be accommodated, such transfers can be routed to an alternate user device set for further action by the user, or simply acted on at a head-end server. In this way the simplified methods can effectively be used to provide a limited-function ITV service, and one that can be configured to operate on its own, essentially as a message service, independent of any more advanced browsing functions.

It should be noted that in the case of commercial programming or advertisements, an additional level of variation in program identity could be involved, in that the relation between advertisements and surrounding programs could be complex. Not only might a given commercial be used with many different surrounding programs (and at many times), but personalization and targeting might cause alternative commercials to be seen by different viewers of the same program. A variety of methods can be used to ensure that interactions associated with such commercials are handled appropriately, for example by including identifiers for both the specific commercial and the surrounding program, as well as other identification of the commercial avail slot, as part of the state information transferred, and by structuring further interactions to use that information to control the response, and possibly to forward that information, optionally with other profile information, demographic information and the like, for use with any remote transaction or other support or monitoring activities.

Again, depending on the particular embodiment, all such state information might be transferred directly from the TV/STB, using whatever path is available, or some state information, as well as other supplementary information, could be included in the transfer record as part of a relay mediated by head-end servers. Thus, for example, even in the case of targeted ads, rich interactions could be supported by using software at a head-end to coordinate with a PC, with little or no modification to installed STBs.

User Controlled Targeting During Browsing

User controlled targeting at the time of any link traversal may be desirable, as it allows the user fully dynamic control in adapting the presentation experience to the task at hand, as it changes through a complex session of interaction. The essence of non-linear hypermedia is that they branch in accord with the non-linear nature of human thinking and task flow. Needs and preferences may change from moment to moment, depending on the path taken and on external factors.

Thus, in TV-centric use, a user starting from a TV program may be signaled that relevant enhancements are available for use if so desired, or the user may spontaneously seek materials related to a program. As ITV becomes widespread, signals may not be needed to indicate general availability of enhancements, but may be desirable to flag items of special interest. Such signals may be a visual bug, as used in WINK, other visual indicators on the screen or on a device within view (such as a set-top box), sounds, or any other signal.

In any case, the user may take various paths from a TV program. He may have a quick casual interest and be content to interact with the TV system in classic one-box ITV mode, without bothering to use any other device sets. However, he may have a more intensive need,

prefer richer function, or want to use a separate device set to avoid disturbing others who are also watching the TV. In that case enhancement content should be directed to an alternate device set. A likely hybrid case may be to trigger some initial interactions on the TV, such as to see a menu of current options, possibly check some EPG information or similar basic and readily requested information, and then perhaps finish or instead decide to go deeper on the PC device set. Thus desired features are to target from pure video to either the TV or the PC, and to target from enhancements on the TV to the PC. In either case, it may sometimes be desired to traverse to a linked resource on the changed device set, such as to go deeper, or to transfer the current resource to the alternate presentation device set, such as to interact more richly with it. For example, in a t-commerce application, the user may be about to order merchandise, and decide that the PC offers better function for filling out an order form for a complex item (such as a new PC).

In such usage with enhancements to a TV-centric browsing experience, it might normally be preferred that the base TV program be considered as a linked session distinct from the enhancement session, so that transfer is understood to transfer the enhancement session, but not the base TV session.

Activation of a transfer by the user may take a number of forms. One key variation is whether the transfer consists of the link actuation that opens a new destination resource, or a cloning that duplicates the current resource in the new setting, effectively cloning the existing session at the target system. This corresponds to the difference between shift-clicking a link on a Web browser (such as MSIE) to open the link ending resource in the new window, and using the File/new/new window menu selection to instantiate a new window with the same cloned resource as the current window.

An additional capability for TV-oriented viewing with a MMUI configuration, and readily provided using the methods described, is to offer some simple commands for swapping and altering video views using PIP (or embedded window) support. A simple command could be a PIP-flip, which for example, could start from device set A with resource 1 in full screen and resource 2 in a PIP, and causes a transfer or swap to device set B with resource 2 in full screen and resource 1 in a PIP. Variations could cause the PIP on device set A to close, and/or omit the PIP on device set B. A simple control command language or macro facility could allow addition of other similar functions by vendors or users.

A wide variety of specific UI controls can be used to provide these and other similar functions, both as standard features, and as user variable options, as discussed further below.

Link Attribute Controlled Targeting For Authoring/Production

While it may be generally desirable that the user have ultimate control of his loci of work and use of device sets and the corresponding targeting of resources, it is perhaps best left to the content creator to determine or at least communicate to the browser a recommended targeting. This may be useful to control details of targeting of individual resources within the constraints of the device sets that have been put into active use by the user, and can also be useful for activating available device sets as well. Embodiments may support both user and author-driven modes of control, as well as means for setting preferences and priorities to govern how such controls interact.

Insight into how link attributes may be used and interact with user controls can be gained from current Web browsing usage, as well as from other hypermedia systems, but broad use of ITV will likely lead to further issues and enriched features, which may be handled by

straightforward extensions of the capabilities outlined here that will be apparent to one skilled in the art.

Reviewing existing practice, HTML has long had basic features for targeting windows and frames within windows. HTML 4 provides for the attribute "target = frame-target" to specify the name of a frame where a document (resource) is to be opened, which can be used in elements that create links, image maps, and forms. Frames may represent an entire window or a designated region within a window. Names can be assigned to frames via the "name" attribute. A set of reserved target names provides for generic controls, such as, for HTML 4.01:

- _blank: The user agent should load the designated document in a new, unnamed window.
- _self: The user agent should load the document in the same frame as the element that refers to this target.
- _parent: The user agent should load the document into the immediate FRAMESET parent of the current frame. This value is equivalent to _self if the current frame has no parent.
- _top: The user agent should load the document into the full, original window (thus canceling all other frames). This value is equivalent to _self if the current frame has no parent.

XLink provides for similar targeting behaviors, but with somewhat different coding conventions and terminology. Here, the "show" attribute has functions corresponding to the HTML target attribute, specifying the desired presentation of the ending resource on traversal from the starting resource. Values currently provided for include:

- new: An application traversing to the ending resource should load it in a new window, frame, pane, or other relevant presentation context. This is similar to HTML _blank
- replace: An application traversing to the ending resource should load the resource in the same window, frame, pane, or other relevant presentation context in which the starting resource was loaded. This is similar to HTML _self
- embed: An application traversing to the ending resource should load its presentation in place of the presentation of the starting resource. This is similar to an HTML alt= attribute, as used with an image IMG SRC link

SMIL provides for similar attributes, generally consistent with XLink, and the related XHTML standard. The SMIL 2.0 Linking Modules attributes include:

- sourcePlaystate, to control temporal behavior for the originating presentation when a link is traversed, with attributes of play, pause, and stop
- destinationPlaystate to control temporal behavior for the originating presentation when a link is traversed, with attributes of play and pause.
- show, much as for XLink (and serving to set defaults or override the sourcePlaystate attribute)

For SMIL target, "This attribute defines either the existing display environment in which the link should be opened (e.g., a SMIL region, an HTML frame or another named window), or triggers the creation of a new display environment with the given name. Its value is the identifier of the display environment. If no currently active display environment has this identifier, a new

display environment is opened and assigned the identifier of the target." SMIL also provides for an area element that can associate a link with a spatial portion of an object, as in HTML, and with a temporal portion of an object (a time span).

None of these existing methods of using attributes have comprehended the proposed extension to multiple device sets, but based on the teachings provided herein, the details of providing such extensions will become apparent to one skilled in the art. Target attributes and special reserved names such as these are readily reinterpreted and extended to work with windows on multiple named device sets. Frame target names can be used to refer specifically to frames (or other similar designations) on multiple displays, and such names could be specified as a simple name, or in a display-frame hierarchy, such as for example in the form of "display-target.frame-target." Alternatively, an additional attribute could be provided, such as "display =" to be used in combination with the target attribute. The special HTML or XLink reserved names might be used largely as is, with the addition of new ones such as, for example, following the HTML model:

- _altblank: The user agent should load the designated resource in a new, unnamed window (or more completely, a SMIL region, an HTML frame or another named window) on the alternate display (whichever that is, based on some set of rules for determining which is the alternate display if more than two displays are enabled).
- _refblank: The user agent should load the designated resource in a new, unnamed window (or more completely, a SMIL region, an HTML frame or another named window) on the referenced, (specifically) named display.

Alternatively, a separate new displaytarget attribute could be defined, with a separate display-target name value, and with separate but similar special reserved names.

Additional special reserved names might be provided to specify alternative displays in a generic manner, such as, for example:

- _TV: The user agent should load the designated resource on the TV display
- _PC: The user agent should load the designated resource on the PC display
- _tablet: The user agent should load the designated resource on the tablet display
- _PDA: The user agent should load the designated resource on the PDA display
- _RC: The user agent should load the designated resource on the Remote Control display
- _computer: The user agent should load the designated resource on the whatever computer-like display (if any) is the alternate to the TV

With such names, it may also be desirable that precedence rules provide alternate mappings, so that, for example, all map to a single display if there is only one enabled, _PC and _tablet map first to one another, and _PDA and _RC map first to one another, and that all of the non-TV names map to _computer if their preferred mappings are not enabled. A list of values might also be permitted, such as _PC, _PDA, _RC, to indicate a priority sequence depending on which of several devices may be available.

Embodiments may be standardized through widely supported bodies such as W3C or others, and the details of coding conventions of the kind just described (or equivalents) could be determined in accord with the applicable standardization process.

With regard to the issue of whether to automatically activate device sets that may not already be active, it may be preferable to consider device sets as "enabled" if they have been defined as being available for use in the current device set group, even if they are not powered on or fully active at the time. In this case, additional attribute codings may be used to specify whether a device set that is enabled but not active is to be activated as a result of targeting.

As noted above, resource coding might exploit advanced markup in XHTML or similar languages to explicitly provide for variant presentation styles to different device sets and form factors. Certain embodiments might also provide coding structures for different device sets and form factors to be used concurrently, with coordinated inputs and outputs, and sensitivity to relevant events, such as based on the DOM event model, with selection of presentation styles made accordingly, such as using CSS, XSL, XSLT, or RML. Such methods may also provide the ability to detect whether a given class of device set is available or active, or becomes active, and alter the specified presentation of the resource accordingly.

Preferred embodiments could also support coding much like the "actuate" attribute that XLink provides for to communicate the desired timing of traversal from the starting resource to the ending resource. This includes attribute values of "onLoad" to indicate that traversal should be immediate on loading of the starting resource, as is the case for an HTML IMG SRC link, and "onRequest" to indicate that traversal occurs only on an explicit post-load triggering event, such as clicking on a link, or a timer countdown, such as for a redirect.

A further feature is to support a link coding that has no ending resource (or equivalently, one in which the ending resource was the same as the starting resource, or equivalently, the special reserved resource name "self"). Such a link might be equivalent to opening a new window with the current resource (and associated context). This feature had little motivation on

a single display, but is useful in a MMUI to establish a session at a second device set, and could be used in resource coding when such action was to be recommended. One example of a use where this could be desirable is when a video sequence that contains spatial hot spots was beginning, so that an enabled tablet could be placed into session with the same video resource in order to facilitate precise selection of desired objects within the image.

It should be noted that alternative methods of specifying MMUI affinities may also be desirable in some applications. For example, instead of the method just described for embedding markup within resources, an alternative is to define a structure of resource categories or types, with specified device set targeting affinities for any resource of a given category or type. This is similar to the simple, highly structured multimedia menu and navigation schemes that were widely used for pre-Web interactive systems (and now used for some ITV systems), and while this may be less general, flexible and powerful than a markup based scheme, it may be simpler to implement and apply. For example, such a scheme might define a navigation hierarchy having base video, interactive indicator bugs, first level menus, i-th level menus, brief text pages, long and multi-page text pages, embedded video, links to embedded video, and links back to primary video, and special-purpose application screens, such as for chat or forms entry, with specifications that might have the first levels default to the TV display, and further levels default to the alternate display, possibly with links to primary video defaulting back to the TV. A table of screen associations can be used to relate types of resources to devices sets. In usage, this could be loosely analogous to the current system tables that define application associations to file types, allowing files to be opened by double-clicking them, and could add screen affinity as a similar association. The coding of such types might be standardized in accord with the MIME (Multipurpose Internet Mail Extensions) protocol, and might be defined as types, subtypes, or

parameters, or it might be defined in accord with MPEG-7 using its Description Definition Language (DDL).

Priorities and Preferences For Managing Device Set Use

The methods described provide for a blend of user and author involvement in determining the targeting of resources for presentation, and as noted, a system for setting preferences and priorities could be desirable. This may take a wide range of forms, depending on the richness of the embodiment and the nature of the balance between author and user that is sought. Details may also depend on the range of content types, content sources, form factors, tasks, and users being addressed, and may evolve as users become adept at using increasingly advanced features.

The structure for setting user preferences may be expressive as to what kinds of tasks should be done on what device, with what rules for transfers and choice of alternative UI configurations. This structure may provide defaults, wizards to allow custom configurations to be chosen based on a few selections, advanced customization and personalization of fine details, and use of multiple settings to allow different presets to be used in different contexts, and support multiple individual users, each with their own settings. The systems could apply settings based on defined conditions, but let the user override them at any time.

A standard structure for setting user preferences could be defined and implemented across a wide range of browsers for consistent use by any system participating in a MMUI browsing experience. Such a structure could start with the defined methods for author targeting in the coding of resources, and for user control at actuation time, such as described above, and could add a preference structure that allows the user to define preferences as to when and how to

moderate or override the coded (author suggested) attributes with regard to specific attributes, device sets, and other usage contexts. This structure could also provide for UI controls to be used at link activation time to control targeting, and specify what level of user controls to enable with what default settings and when they may or may not override the resource codings and preferences.

A simple example of mixed user/author control in current browsers that is suggestive of desired expansions is the use of new windows in current versions of MSIE with current HTML. The default for a simple link (traversed with the standard left-click action) is to open in the current window. A user may force a link to open in a new window by shift-clicking the link, or by right-clicking and selecting the "open in new window" option from the pop-up list. This new window feature is popular for viewing resources from a list of links, such a search results page or menu, while leaving the list in place in the original window. An author can code a link to automatically open in a new window (using target=_blank), and this is popular for cases of loosely associated resources, such as references, or for links to another site, when the author wants the current resource to remain available. Currently the coordination of these methods is not well provided for, and it may be difficult for a user to override a coded attribute, or for an author to be adaptive to user controls, and this is compounded by the fact that the author coding is not transparent to the user, so the targeting to a target=_blank may come as a surprise to the user. Some limited override capability may be available in special controls (such as dragging a target-_blank link to the browser address entry box to open it in the current window). These simplistic controls, and the lack of transparency into coded attributes results in conflicts between site authors and users with different tastes, and no clearly desirable solution.

Addition of features to set preferences and priorities by the user, and to make coded attribute behaviors visible (such as by using alternate cursor shapes, colors, or other effects) could be desirable to make such behavior more predictable and controllable. Thus a basic structure could relate the following for each link attribute behavior coding

- what intent was suggested by the coding
- what visible UI cues prior to activation was suggested by the coding
- what actual behavior should be taken based on a user preference
- what actuation-time overrides could be make, using what controls

Further capabilities can be built upon such a preference and priority structure to allow preferences and priorities to be defined as named sets that apply under specified conditions and also to be manually activated or deactivated as desired. These could be set in two major dimensions: 1) the dimension of device set groups, and what options apply with which groups, and 2) the dimension of task modes, where different behaviors are defined for different task modes. Task modes might be defined to relate to general activity types, such as casual, intensive, multi-tasking, and the like, where the user sets the activity type that is operative at any stage of a session, and changes that as desired. Task modes might also be defined by convention and set globally by authors/producers, using attribute codes, to set modes by content type, which might be similar to task type, or more specific, such as extended content, annotations, references, or special applications, or to set modes corresponding to navigation levels, such as menus, content, subordinate content, and special applications. Other sub-dimensions of task mode might include private versus shared, utility versus entertainment, and the like. Settings might also be variable with respect to different device sets or device set groups, both in terms of form factor,

and such factors as location, such as living room versus bedroom versus kitchen versus auto versus walking versus sitting outside the home. Given the number of dimensions and variables possible, this could be simplified by building in standard default schemes, with simple alternative schemes pre-set as default or selectable as a unit for average users, and with advanced structures accessible for change at a fine level by advanced users (or their support providers). A further variation in controls can be provided to allow users to override markup suggestions (or other produced-in schemes) on a one-time, session, or permanent basis. Some embodiments might also add various learning techniques to gather knowledge of user behavior and apply artificial intelligence and related inference techniques to infer the user's desires. Such behavior controls can be obtained and composed from a combination of sources, including the user, the author (including the full range of sourcing and distribution players), and from hardware and software vendors, third party services, and other users.

In the case of an alternative embodiment of targeting controls using an explicit category structure for resource types and affinities that is based on the types set at time of authoring, as described above, user controls operable at time of viewing might be similar to those described earlier, and a similar preference override feature could be applied to change such target affinities on a type-by-type basis.

Source and Destination Session Behaviors and Synchronized Tracking Options

As noted previously, there are issues of what behavior is desired at each device set after a transfer, and these behaviors can be defined by a similar combination of defaults and preferences, author coding, and user selection. Basic alternatives may be specified for whether sessions end or pause or continue, whether some or all windows close, the disposition of related

UI elements (such as parent frames, for example), and whether device sets are deactivated or powered off. As with other control options, these alternatives may be defaulted by a system, set as user preferences, coded by author/producers using formats similar to the SMIL sourcePlaystate and destinationPlaystate attributes, and controlled by the user at the time of actuation.

These behaviors could take a different shape if the optional tracking/synchronization feature is provided and selected. In this case, the source session might normally remain active and be kept in synchronization with the destination session, which might be instantiated with the same resources active, and with equivalent session state. In the usual case, user action might be permitted at either device set, and could trigger an event transfer with the essential state information to replicate the effect of that event at the other device set, just as if the user action had occurred there. Alternative forms of tracking may be provided, as options, to make one of the device sets operate in read-only fashion, so that it may receive UI events from the other but not originate UI events. Certain embodiments might also provide for temporary disconnection of a device set from active tracking, but with the capability to save events at either location for later re-synchronization when the device set reconnects, similar to the analogous capabilities in (and possibly building on the facilities provided in) advanced collaboration systems such as GROOVE.

Tracking support can be implemented on a single user or multi-user basis. Multi-user support is typical of conventional collaboration or groupware systems, and requires additional attention to managing user identity, security, privacy, privileges, and the like, with regard to both an individual user and the membership of multiple users in an active collaboration tracking session. It will be understood that groupware collaboration is directed to this issue of multiple

users, and conventionally assumes that each user has his own SMUI system (typically a PC, and typically at different locations), with the objective of giving the multiple users the illusion of shared access to a single system that reflects the actions of any of them. Such systems must maintain full synchronous replication of all UI input and output events to each participating system, which is not what is generally desired when a single user uses multiple device sets (and may wish to allocate tasks or sub-sessions to a single device set, as extensively described above). As will be apparent to one skilled in the art, the collaboration tracking/synchronization methods used in conventional SMUI groupware systems are readily adapted to extend the single user, MMUI tracking methods that have been the primary subject herein to the proposed further inclusion of multi-user, MMUI collaboration tracking as a complementary mode of use. In drawing on groupware art to extend it to the objectives of a multi-user MMUI, it is useful to recognize that a range of methods have been used in groupware relating to different coordination levels of display (image) broadcast versus event broadcast, the use of centralized versus replicated state control, and whether the applications are collaboration-aware, and these issues have parallels with the MMUI architectural variations described herein. Recapping the teachings provided herein with regard to the variations just noted, the objectives of browsing can be well met in most common cases with methods that are collaboration-aware and MMUI-aware, and with the granularity achievable with selective event replication. These may be applied in either centralized or replicated state control contexts, depending on the platforms to be supported. Nevertheless, MMUI techniques could also be accomplished by display replication, and software environments could be provided to permit applications to be MMUI unaware.

It should be noted that some support of management of user identity, security, privacy, privileges, and the like may be desirable even without multi-user collaboration features. This would protect the security of individual users, and allow for multiple independent users of the systems supporting the MMUI to set individual preferences, maintain individual history lists and other individual context information that could persist across sessions and be protected from interference by other intervening users, such as other family members or office mates. It should also be noted that a further useful feature in a multi-user case might be to transfer a browsing session to another user, whether at an alternative device set, or using the same device set. The former could be useful to pass activity to another TV viewer who has his own personal device set (whether collocated or not), and the latter could be useful to change the context of a session to take on attributes and privileges associated with the second user, such as to conduct a transaction. The details of supporting such controls and transfer features will be apparent to one skilled in the art based on the teachings herein.

Congruent Behavior Of Independent, Dependent, and Centralized Systems

Described in the foregoing discussion has been the case of a true MMUI, where the coordinated device sets that compose a MMUI are independent systems, being driven by separate processors. From a user and application perspective this is just one end of the broader MMUI (or MDUI) spectrum from centralized systems to partially interdependent systems to fully independent systems. From a technical implementation perspective, it is the one that is most challenging, given the need to transport and reestablish all relevant aspects of session state. Most of the teachings provided herein for that case are also applicable to the more centralized

and interdependent cases, but embodiments for those cases can be greatly simplified by exploiting common elements and common access to state information.

Examples of alternate embodiments of a MMUI with variant, more centralized architectures are possible configurations 1) of an advanced TV, with PC-like capability and able to directly drive a tablet, laptop-like or PDA-like I/O device set, or smart remote control, and 2) of a PC-DTV system that supports an across-the-room TV monitor and remote control in addition to the usual PC I/O device set. These are fundamentally very similar, but can be expected to differ in that they come from very different hardware, software, infrastructure, and business "cultures," that shape then quite differently. Other architectures that may be more or less centralized may relate to more PC-centric assemblages, such as Pebbles-style PC-PDA combinations, Mira-style combinations of tablet with PC, assemblages that loosely couple TV STBs with separate home entertainment controllers, video game systems and many others.

Externally, it may be desirable that the MDUI behave consistently for browsing and transfer of work locus regardless of the system architecture (true MMUI, centralized, or whatever), effectively masking such internal implementation details from the user, except in so far as attention is needed to set up and manage configurations, device set groups, and such. It may also be desirable that the user be generally unaware of whether he is browsing across a system with one processor and software image, or one with many.

Internally, with regard to software implementation, much simplification is possible in the case of a centralized system in which all device sets are driven by a single control processor. In this case it is possible for all browsing to be done with a single browser software instance, with direct access to all browsing state and context information for all of the device sets in use, and with true common access to the hypermedia storage layer, as well as to all system storage,

including persistent storage and transient storage used for resource caches and system caches and other transient state information. Thus much of the import/export/tacking function is obviated. Instead of transferring sessions and state, the essential task reduces to pure coordination of sessions and state across device sets, which depending on embodiment, may be generally similar in nature to that of coordinating browsing with multiple windows on a single device set.

It should be noted that such centralized session processing and control limits the overall function provided in a number of ways. Ability to add and remove a device set from coordinated use may be limited. The central system must be capable of recognizing and driving all desired device sets at a level that is sensitive to their form factor and rendering capabilities. Thus, for example, whether based on a PC or STB, the centralized browser must be equally at ease with the details of device control and of rendering presentation resources for a high-resolution PC screen, a TV (SD or HD), a PDA, or a display-equipped remote control. In general, the openness of such architectures to systems and device sets of widely varying architecture and configuration, provisioned from varying sources, and assembled and applied in an ad hoc fashion is more limited and difficult to adapt to unexpected requirements. With the decentralized methods described herein, all that is needed for an arbitrary system to participate is an exporter/importer that is capable of transferring session state in the standard, high-level format and granularity established for any member of a MMUI device set group.

Intermediate cases may be addressed by these methods as well, and depending on their architecture, more or less of the transfer import/export process could be required to achieve coordination. Systems may be technically independent in that they are separable and use independent processors, but for purposes of coordinating a MMUI, they may run in a centralized application mode. Common collaboration systems like MICROSOFT NetMeeting and others

based on ITU T.120, run in this mode, as do some thin-client terminal systems like Windows Terminal System. In this case, the device set displays are not driven by local applications, such as browsers, that render onto them locally, but by display replication from a central system that runs one central copy of the application that does the rendering. Coordination across device sets is at the level of direct UI input/output actions, not at the level of sessions. There is really just one application session. Thus many of the constraints of a centralized system generally apply to this architecture as well.

A similar variation that exists at a software level is also to be noted. A single controller system processor, or multiple processors in a tightly coupled, shared memory, multi-processor system or a cluster or other assemblage such as a grid system presenting a single system image may run multiple independent browser instances (running as separate software processes) that share the operational logic of the processor(s) and other basic system resources, with each instance controlling separate device sets, and each capable of coordinated use together, much as for independent browsers. A very simple example is that of running multiple instances of a browser, such as running two instances of MSIE on a Windows PC. In such a case, some aspects of the context may potentially be commonly accessible, such as page resource caches, history lists, cookies, and storage systems, but details of session state (such as current page, navigation path, window configurations, and forms data entry state) may be local to each browser instance. Thus MDUI coordination might not be supported across those browser instances without the addition of the methods for state transfer described herein, simplified somewhat by the common hardware, software and context elements in ways that will be apparent to one skilled in the art (such elements may not need to be transferred, or may be transferred by reference only). Another example might be a case of running multiple different browsers on the same system,

such as might be done for an advanced TV with PC functions or a PC-DTV system that might run a conventional ITV browser for enhancements oriented to the TV, and MSIE or Netscape for Web browsing. Here again, coordination might not be possible without adding support for the coordination methods described herein, and here the simplifications relate primarily to the simplifications of communication within the integrated hardware environment.

To complete this discussion of independence, it should be noted that there is a more extreme case of independence, that in which common access to the hypermedia resource storage layer is not available or readily used. This may occur in cases where network access is constrained, and devices may have only limited capability for signaling one another. A related variation is where resources are distributed on physical media (that are mounted locally) and separate access is more readily provided than shared access. In such cases, the transporter and state records could be expanded to add context details on the resources in use sufficient to synchronize or transfer desired portions of the resource content along with the session that presents them. One example might be that of enhancements to a video (or music, or other) resource that is on CD-ROM or DVD, where one device set may have a copy of the stored resource, and the alternate device set may have a duplicate copy, or a server-based copy, or just the associated enhancement resources.

Transactions, Cookies, Peer/Server State, and Editing

Further issues in coordinating MMUI sessions apply in cases where state is complicated by transactions and related data entry, cookies, and state for an associated server or peer system session. Many similar issues apply when editing of hypermedia resources is supported. Methods for addressing these issues are described in this section.

For both transactions and editing, additional details of state related to user input to transaction entry forms, such as for example HTML forms (or XForms), or to an edit-capable browser or equivalent program, must be maintained in the state record to enable a finely grained session transfer capability. This relates to the relatively fine granularity of data entry input and control interactions, as opposed to the more coarse granularity of link traversal in simple browsing. The simple solution is of course to not support this finer granularity, and to have transfers ignore any such entries if not submitted to the server or committed as completed edits to a resource in the storage layer. Such a solution may be quite satisfactory in many uses, and users could adapt to avoid intermediate transfer attempts, or live with the task of re-establishing the entry details that were lost. This limited level of support is consistent with how some (but not all) current Web forms processing handles the back button, where the form is presented in its original empty state, and any user entries are lost.

Naturally, more complete support may be desired by users, and this can be enabled by including the edit state details in the state record when exporting and importing a session. Specific methods for this may vary with the details of browser or editor implementation, as will be apparent to one skilled in the art, but the basic result is a structured list of all edit entries, including such metadata as needed to identify the elements and positions to which such entries refer. In the case of form fields, this could list entries by field name and value, or, as in the case of XForms, provide an XML structure for the form entries and values. In the case of resource editing, such details might be a trace of changes similar to that maintained by a word processor for use in undo/redo commands, and for recovering changes after a crash (including replaying all intermediate events, using snapshots of state, and hybrids of the two). Emerging standards for access to resource structure and manipulation events, such as DOM and the DOM event model

may be used to capture such finer-grained state elements. This would permit access to such events as entry to and completion of a form, and to the filling of fields within a form. Other finer-grained state details may also be captured by similar means, including well-known interaction points in browser interaction widgets, and can thus be used for more fine-grained coordination. Methods for forms input capture are similar to those used in browser form-filler programs and autocomplete features. A Microsoft facility, IPersistHistory, enables the saving of forms state data for use at a later time on the same machine, and similar techniques can be adapted to extract this data for export. For example, this data can be obtained from (and inserted into) the Dynamic HTML Object Model used on IE 4.0 and later browsers, using the IHTMLFormElement Interface, and other browsers offer similar access to the document object model (DOM).

Also a factor in finer-grained local state is the use of embedded logic and of plug-ins or other support logic that extends standard browser functions. The simplest solution, again, is to set a granularity that ignores the internal state of such elements (such as by starting them over), and this may be quite satisfactory for many applications. A more complete solution is to address granularity at the level of user inputs, in which case methods similar to those described for forms and editing can be used, including tracking and replaying all intermediate events, using snapshots of state, and hybrids of the two. Capture of full software process state is also possible, but much as for the browser itself, would involve a level of complexity that may not be warranted.

Regardless of the level of local granularity desired, transactions (including queries, e-commerce, and the like) and many other kinds of remote sessions maintained between user system and a remote peer or server system are usually stateful, and require support if that user

session is to be transferred to another device set. Such situations can be understood as involving a user session that at some point enters into a remote session with the peer or server, such as to place an order or to obtain services that involve a continuing session identity over a series of interactions. A given user session may over time have multiple (sequential or overlapping) remote sessions associated with it. Thus to transfer a session, while it participates in a remote session in a way that allows the remote session to be maintained, requires that the relevant state information for that remote session effectively be transferred, and that the transferred local session be able use that information to re-establish and join with the corresponding remote session. In the case of a centralized browsing system and single browser instance, this is relatively simple, since most of the change is invisible to the remote system, but for independent browsing systems it is more challenging, since the network address and other system-specific information (security certificates, system identifiers, and cookies, associations with shopping cart information and other transient or intermediate transaction state, and the like) for the new system may be changed. Similar issues and methods apply to client-side wallets, passports, or other identification, profiling, and preference data.

An intermediate, somewhat simplified, case occurs when the transfer occurs at the point of actuation of a traversal that begins a remote session. In this case, the transfer process need only transport and import all persistent cookies (or other persistent state information) that may be associated with the domain of the specified server for the target system to be able to initiate a session with that cookie information.

The more difficult case is that of a transfer after the session has been initiated from the first user system. One method of handling this case is to hide the change in address from the remote system, and this can be done by relaying through a proxy server that is set up to mask the

true address of the user system. Such a proxy would preferably be secure and trusted to protect against unauthorized or spoofed session transfers. This proxy could be located at the first user system, and that would offer the benefit of enabling the proxying to be started mid-session on an ad hoc, as needed basis, using the already-established network address. More generally, such a proxy-like server can be embodied in any local controlling devices, such as a gateway, set-top box, or any PC, to allow multiple coordinated devices in the home to appear as a single device and single client IP address to the remote server, again retaining transparency at the server. In the case the control device would replicate the cookie or other server-visible state information just like the other (local) state data, or could maintain the cookie (or the like) at the proxy (in a manner similar to that used in proxy servers that support wireless devices).

Similar proxy functions can also be provided by a remote server, whether an independent proxy server, or integrated with a remote service. One alternative approach is to add support to the remote peer or server system to enable it to be aware of the session movement and to adapt to the changed network address and system identity. This might take the form of protocol-level support at both the client and remotely to deal with replication or migration of a client that is treated as a single user. This could also be done in a manner generally similar to support for roving IP addresses that is intended to support mobile devices in Mobile IP, based on remote proxy-like approaches in the network (in that case at the wireless access node, and thus transparent to the server), and in more advanced context-aware portals. Such support could also draw on the somewhat related methods used in Windows "roaming" support to allow a user's basic (static) Windows system preferences to be passed to a server and used to personalize a second PC to the same settings. In this case, the state replication logic described earlier might maintain the distinct identities of any network-visible and network-addressable client

components. This can be further supported by the browser transfer process by sending a transfer notice to the remote system stating the details of the new systems address and identity, and again preferably providing security authentication information to protect against unauthorized or spoofed transfers.

Server sessions may be particularly critical to support of ITV services that rely on head-end server support to provide functions supplementary to those of limited-function STBs, such as analog STBs or basic digital STB models (such as, for example, MOTOROLA DCT2000 or SCIENTIFIC ATLANTA Explorer 2000). This may include cases where a many or most interactive functions are controlled by the head-end server, with the STB acting essentially as a thin client I/O controller that is architecturally similar to WTS, CITRIX, or X Windows (even including remote PC-like functions such as using the PEACH NETWORKS technology acquired by MICROSOFT). In such a case, coordination could be managed at the server, using the methods described in regard to the STB, but with alteration of the communication paths to include the wide-area communications required. Such communication could be routed from the server to the STB on the cable or satellite facilities, and then locally from the STB to the PC or other coordinated device set, but might more readily be routed from the head-end on an alternate path to the PC, such as via the Internet. This method of using the TV/STB as a thin client-based device set, with support at the server for coordination with a PC-type device set could be very attractive as a way to provide rich coordination using current installed base hardware, with little or no change to STB software, with most or all of the added software at the head-end (linking over the Internet to enhanced browsing software at the PC). This is a variation on the embodiment described earlier in regard to state transfer via head-end and Internet. Further variations could add flexibility to offload intelligent function from the head-end to coordinated

PCs, including such functions as basic interaction navigation support tasks, more advanced browser functions, such as interpretation of image region hot spot metadata as link arcs (similar to that done at the head-end by VEON servers), and specialized and supplementary services. A further extension of this approach could place all coordination and state management intelligence at a remote server, acting as a central controller, so that all local device sets operate as (more or less) stateless thin client/terminal systems. Such thin client/terminal function could be at the level of display image replication, or at a level of simple browser rendering and basic navigation functions. More generally, this method permits a spectrum of embodiments that vary in terms of how coordination function is distributed among TV-type system elements and PC/PDA-type system elements at the user location, and head-end or other remote server-based elements, including mixed cases supporting multiple distribution architectures. Such methods can be used to build a range of extensions from current 2-box ETV offerings that offer greater integration and coordination of the two boxes, potentially including the high levels of flexibility in targeting enhancements to the TV or the PC/PDA described above, and that can work with installed-base STBs as well as more capable units. It will also be apparent that such methods of placing coordination functions at the server may also be applied for coordinating state for server-based functions (including transaction functions) in general, as an alternate to the more peer-to-peer and distributed methods described above, and that such methods may be particularly attractive in contexts where the necessary reliance on remote services is not viewed as a disadvantage. Note that in such cases, the Internet path could happen to be physically on the came cable as the TV signal, such as by using a DOCSIS Internet connection, but be logically distinct.

State Tracking Simplifications

The methods described herein for state transfer at varying levels of granularity offer significant advantages in simplifying the coordination task. This can be understood as exploiting strategies for limiting when events are tracked and how they are collected and communicated. Existing and proposed systems for collaboration and systems for simultaneous multimodal interaction generally require ongoing collection and broadcast of state-change events in realtime while such activity is active, and collection and storage of such events for later synchronization if delayed activation of such features is to be permitted. In the case addressed herein of session transfers, this support can be selective as to what is collected and what is exported. The methods described herein seek to reduce the portion of state exported to the minimum needed for a given function.

This was presented in terms of ad hoc batch export/import of state, where relevant state is extracted from the current model (or the internal state of the browser) at the time of the transfer request, but the same strategies can be selectively applied to event-oriented methods, like those used in collaboration and synchronized sessions, as well. Instead of extracting state from the model, event-oriented methods can be thought of as tracking events that cause changes to the model as they occur, before they cause the change to the model, such as in a mediated model-view-controller structure, and then replaying the events to replicate the corresponding changes to the state of the model. To further clarify that simplification process in terms of such an event-tracking embodiment, the specifics of what event-related state details need be retained and transferred depend on both the nature of the activity and the granularity of coordination desired. Simple logic in the browser or exporter can monitor events, and discard those not needed (such as for being out of scope). This simplifies event tracking, export, and import. For coarse

granularity, events may be at the level of resources visited, loaded and/or presented) Current state could have no history at all, but could have some defined range of history, so that, for example, entries within some defined number and path distance could be saved, and any excess could be deleted. For finer-granularity, edits, forms entries, and intra-resource navigation events could be recorded, but once there is a commitment to the storage layer, or a traverse to a resource that replaces the subject resource, some or all of those may become irrelevant. Alternative embodiments may flush all such events, or reduce them to a minimum end-state set that removes those that are redundant or reflective of a transient intermediate state. These simplifications may also involve combining events into new but equivalent composite events that consolidate the effect of multiple intermediate and partial actions. Thus for example, a final set of forms inputs might be retained (to support backtracking), but not any intermediate actions that were altered. In MVC terms, this can be thought of as maintaining a secondary, parallel model, one specific to modeling only those aspects and granularities of state relevant to transfers to other models. Such methods can enable considerable economy in retention, transmission, and import of state, particularly if a possible transfer or synchronization is to be enabled long before it is actually requested, and can make it feasible to have such a capability active and in place at all times, unlike less efficient methods that might tend to be used more selectively, and thus might not always be ready for use when desired.

Because of that, these methods are also useful to parse out redundant or obsoleted events in the case of a collaboration or a simultaneous multimodal application, in those cases that realtime synchronization is not active and some set of events is held for future use (such as after a connection or reconnection), when synchronization is to be established based on this pending event stream. The same simplifying parsing and deletion functions can be applied, and this may

make wider use of such applications feasible as well. This simplified method can be used in alternation with full synchronization, so that this secondary, simplified model is maintained locally until synchronization is desired, when it is used in an input event re-feed process to bring the systems into initial synchrony, which is then maintained by synchronous event transfers, and then reverted to when synchronization is deactivated. These methods apply both to systems of conventional design, and to those using model-view-controller designs. Expressed in MVC terminology, what these methods do, in at least one aspect, is enable systems to operate with independent models, but to coordinate those models when needed, in a simple, low cost manner, regardless of whether based on a central intermediary, or a purely peer-to-peer process (or some combination). Having independent models can be a significant advantage for systems that may be used separately, in disconnected mode and/or in different applications, thus enabling ad hoc coordinated use. To make such methods robust in the case of a varying number of active and inactive group members (collaboration participants and/or device sets), it might be desirable to determine synch-points when members enter and leave the active group that can be used to define intervals that have no membership change, within which simplifications can safely be made with no risk of affecting a re-activation for a member at an intermediate state that reflects some but not all flushed events.

Also, while the examples of event stream simplification given here are oriented to browsing events, it will be apparent to one skilled in the art that similar event reduction methods can be defined, based on the principles described herein, for event streams relating to many other applications. For example, these methods apply with only minor adaptation to a shared/synchronized notepad/editor application, and quite similarly to a shared/synchronized sketchpad. Similar simplifications for a shared/synchronized calendaring system could eliminate

large numbers of events relating to scheduling of items that are later deleted or moved. Similar methods can be used for contact managers, discussion forums, file management, image management, games, computer-aided design, supply chain management, and other shared/synchronized applications.

Some other methods for simplification of state tracking that may be useful in some embodiments are noted. One method that can be useful, such as when assembly of a full state export is costly in processing, is to use a hybrid of full state exports and event logs. This may be implemented in a manner similar to the checkpoint/restart method used in data processing transaction logging, where a full dump or image copy is made periodically, and an event log is maintained only from the time of the last checkpoint. Another form of hybrid is to supplement the purely current internals of a model by maintaining some aspects of state in a partial event log-like form, which may be useful for aspects of state in which history is relevant, such as for a browsing navigation history. It should also be noted that in synchronizing video to begin viewing at the same point as the originating system, the need for replay at the new system can be avoided by including the current time-position as part of the state, and time-normalization methods can be applied, using calculated delays, reference time servers, and the like, to adjust for transmission latency. Thus where a pure event-based synchronization method might have difficulty positioning a stored video to an intermediate time (without playing through it and incurring delay), current time-position data enables random access to the desired position.

Security, Privacy, and Digital Rights Management

Allowing sessions to be migrated from one system to another may call for attention to extending current and future methods for ensuring security and privacy, and for digital rights

management. Such could allow authorized users to have proper powers on both systems, and to exclude others and to prevent spoofing or other security flaws. Similarly, depending on the specific context, rights and entitlements to digital resources, including conditional access and copy protection, could presumably be provided to a user without regard to the device set used, or alternatively, with only incremental cost for access from a second device set. This may also involve the use of multiple point-of-deployment (POD) modules for hardware-based control of such functions, or an enhanced POD module with support for controlling multiple systems. Specific methods for doing this relating to user identities, device identities, network addresses, and other identification, certification, and authentication methods will be apparent to those skilled in the art.

Hypermedia Edit Applications

While most of the examples discussed have focused on browsing pre-defined content, the same principles should be understood to apply to edit applications, and the value of multiple screens is especially clear there.

Limited multi-screen edit already is already used for specialized video editing systems, with editing tasks on the PC that drive viewing of (pure) video on a directly controlled TV monitor. As personal editing of music, video and stills gains popularity, limited editing functions may be expected to be provided on TV-based home media systems. For example an APPLE iPhoto-like photo viewer/editor may allow showing slide shows on TV. Such a show could be created on the PC, and presented (as read-only) on the TV, but some limited edit could be desirable, such as to change sequence, delete or add images, change timing, etc. This could be done purely on the TV system, but here again, the ability to seamlessly transfer that task to a

coordinated PC could be desirable. That could permit more complex tasks such as searching a library, enhancing photos, etc., and could allow such tasks to be done in real-time by one person during a group viewing that continues without interruption. For example, while showing a presentation of vacation pictures to a new girlfriend on a TV screen, the host might decide some of the more intimate photos of a prior girlfriend should be unobtrusively deleted from the show. This could be done effectively with a coordinated PC/PDA/tablet, if such functions were well coordinated and commonly used. Similar functions could apply to music playlists and compilations, and to video content.

Future hypermedia may also be more readily editable, unlike the current Web, but like other hypermedia systems. In that case, coordination of browsing/edit functions might be even more widely applicable. From this perspective, chat, bulletin board forum services, and Weblogs, and other forms-driven applications, represent a current, limited form of browser-based editing on the Web, and one where the proposed coordination could be beneficial, such as to facilitate chat on the PC/PDA/tablet during a TV viewing. (For example, the coordination can allow seamless activation of chat associated with the currently viewed TV program.).

Venue/Kiosk-Based Coordination

An application of coordination that highlights the value of coordinating loosely coupled devices is the case of a venue that provides systems for coordinated use with customer systems. These may be considered as venue or location-based services that are delivered using a kiosk or other device set provided at the venue.

One example is a hotel, which may have a TV system that provides TV, ITV, and Internet connectivity to guests, in which the guest may connect a portable computer (or PDA) to

a network jack. Current systems provide such services to a PC as simple, uncoupled, network access only. Adding coordination enables the guest's portable computer to be used as a lean-forward device in conjunction with the ITV system. This kind of loose coupling is essential to the separate provisioning/ownership/control/security issues of the guest/venue relationship. Similar services can be provided at airports, Internet cafes, etc. Similar services can also provide full function PC kiosks for ad hoc use with PDAs. Various well known security measures could be used to control what information is sent from the personal system to the kiosk system, to provide secure transient services on the kiosk that are reliably erased after use, and to certify the integrity of identities and such safeguards.

Conversely, the venue-provided TV could be used as an auxiliary display for PC-driven content, such as viewing DVDs or playing games (PC or Internet-based).

Similar applications also relate to the advantages of ad hoc use of independent PC/PDA devices in home or office contexts (as opposed to dedicated, fixed function, multiscreen systems). Other example apps:

- Hotel reservations/guest services/tourist info/T-commerce – video plus ITV on the TV, shifting to the PC for more detail/interaction intensive tasks, and tasks that depend on PC-resident resources
- Education TV plus adjunct Web/applications – view lecture and related content on the TV, get supplements on PC (simulations, references, notes/annotations, and the like)
- Multiplayer games – main shared view on the TV, controls on the PC, if available, and optionally add more players on a bring-your-own PC screen basis

Flexible and Deep Integration With Other Applications

It should be emphasized that coordination functions can extend well beyond browser functions to include data/file transfer, distributed transactions, and Web services between coordinated devices. This adds further value to the use of "loose," ad hoc coordination of independently capable and separable devices.

Services on the more personal device can support integration with personal applications, ranging from simple file downloading and saving from the Web or other hypermedia to deep integration with personal thick-client applications and databases such as INTUIT Quicken and the like. Some examples are listed for the hotel/venue context, above but similar examples apply in home and office environments:

- An example of such integration includes seeing financial news on TV, getting details on ITV (on either the TV or PC), initiating a stock trade that links to an electronic trading service and a PC-based portfolio system (like Quicken), and then using the portfolio analysis system (at the time, such as to evaluate asset allocation, or later). Wide availability of such kiosk/venue-based convenience services might add to the appeal of thick client applications in spite of mobility requirements.
- Another example might be to use DVR and personalized EPG functions (on the PC or TV), based on rich personal preference profiles on the PC, to control use of the TV in a hotel room.

Services on the TV or kiosk device may also benefit from use of the personal device for control of that device and its service functions (such as smart remote control of content viewing),

and in the case of a venue such as a hotel, for venue-related services, including transactions, commerce, and concierge services at a hotel kiosk.

As automation grows, access to such personal applications and databases from any device and for use with any service may become increasingly important. Without coordination of TV and PC devices, it may be impractical to save information from a TV interactive session for use with the PC, except through use of an intermediary network-based storage relay service. Such a service could involve additional steps, and additional security exposures. Similarly, PC data can be more effectively used to better personalize (or be used directly) in the TV session, as for the hotel DVR/EPG example. This applies broadly to uploading or using data resident on the PC, such as personal application data, including personal productivity applications such as contact lists and e-mail and collaboration management, finance and personal business in conjunction with an ITV session.

Any suitable means of distributed application integration may be used, including remote procedure calls or message oriented requests, and Web services using SOAP (Simple Object Access Protocol) and XML (and related service discovery and brokerage services such as UDDI, WSDL, and the like), as well as simple file download/upload. Once serious use of ITV has begun, it may become increasingly valuable to have full integration of ITV-based services with PC-based applications. Application-specific integration methods, such as those using Open Financial Exchange (OFX) and Interactive Financial Exchange (IFX) could also be desirable.

From a broader perspective, it should be understood that these capabilities go beyond pure UI capabilities relating to MMUIs and use of complementary form factors, to include rich distributed processing capabilities that allow multiple systems to be used in browsing, and in

related activities that integrate advanced and/or specialized processing, data access, and service functions from multiple systems into the browsing process.

Virtual and Augmented Reality

As device sets and UIs evolve toward rich use of VR, such as in immersive environments, representations of virtual device sets may become more relevant to UI design and usage than the details of how those representations are implemented, whether using real device sets, or simulations of various input and output artifacts available to the user (such as display screens, viewing regions, keyboards, buttons, haptic devices, cameras, microphones, and the like). In such embodiments, MMUI issues apply to the virtual device sets, as well as the physical ones, and it is coordination of browsing activity across the entire suite of virtual and real device sets to meet the needs of the user that is to be desired. The task of providing a MMUI that allows user control, in some combination with author recommendations, is essentially independent of whether the device sets are real or virtual. The task of implementing browsers to support those MMUI features, whether on centralized or distributed systems, is essentially as described above. Any adaptations required in any specific VR embodiment will be apparent to one skilled in the art.

Revenue Models

According to embodiments of the invention, the MMUI features and services described here may generate revenues in a wide variety of ways, depending in part on whether they are local to a user's site, based on remote services, or venue-based, and on whether they serve the user alone, or facilitate revenue generation indirectly by enabling transactions, commerce,

advertising, or the like. MMUI features can add value to device sets and systems, generating revenue in the form of increments to device prices, operating system or application software prices, or as separate software products. MMUI features and services can also be offered in conjunction with ITV, Web, DVD, or other hypermedia services, whether local or remotely based, and thus enable price increments on a subscription or item-oriented basis. Those features and services can also add to the value other remote services, such as e-commerce or information services, and generate revenue by increasing price or sales volume of those services. Similarly, venue-based services may generate revenue in the form of direct charges based on duration of use or units of service, or indirectly by adding to the value and price (or sales) of related services. Services for transport or relay of session information and related content, and services relating to portals, may also generate revenue in the form of service fees from users, carriage fees from content providers, and advertising or sponsorship fees. Numerous variations on such themes will be apparent to one skilled in the arts of entertainment, e-commerce, e-business, and system/appliance businesses.

Comprehensive Framework For MMUI Browsing

It is suggested further with regard to the methods described here that the whole is far greater than the sum of its parts. In at least one aspect, these methods provide a comprehensive framework for browsing using a rich and flexible MMUI, across a wide spectrum of systems, device sets, hypermedia types, and browsing application forms and tasks. By providing a single common framework, a consistent user interface, shared support facilities, and common infrastructure these methods can be applied in unlimited combinations and extensions to make the use of MMUI browsing feasible in varied contexts and niche applications in which creation

of a less comprehensive, specialized point solution may not be economically feasible, and for which the user learning curve for the concepts, features, and controls of a specialized system could seriously limit acceptance and use. The methods provided here and the UI concepts and metaphors they embody can be broadly applied on a consistent basis, with only a moderate level of adaptation that is of a kind that could be recognized as natural to the context (by both users and developers), and thus readily understood and implemented. While convergence of TV and computers has been much discussed, this work offers the first broad method to achieve meaningful convergence in the user's interface to, and use of, TV and computers.

Stepping back to look at the broad concepts of use of a MMUI, some representative examples of typical useful modes of coordination may clarify some of the many ways the invention can be used. Implementations might offer any or all of these combinations, as well as other variations. These examples are summarized in the table in Appendix A.

Beginning with the section headed Display Set Options, the first column lists the major elements of viewing and interaction along the lines of the examples of FIG 3. The next two columns describe display options on each of the TV and PC/PDA. The last two columns describe the scope of commands and other user interactions from each of the TV remote control and the PC/PDA. Some explanatory notes on the table:

- The PIP elements may be implemented, for example, as either a true hardware PIP in the TV (with two A/V input signals), or as a software-defined window or frame controlled by a video compositor and/or computer.
- Video windows within a browser screen are notable with regard to how the liberator method for access to a broadcast signal, as described above, could enable replacing a standard streaming IP source with a more effective broadcast source.

- The combinations of display and control may be different for each element type, or even for different instances of element types, and standard and customized usage preference profiles might be used to provide for one or more useful combinations.

A typical use may be that a primary video program be viewed on the TV for maximum effect (theatrical immersion), and that interactive text such as browser screen (or secondary video elements) be viewed on the PC/PDA for ease of reading and interaction, and to avoid disturbing viewing of the ongoing TV program by the interactor or others. Varying uses of PIP elements then follow this affinity, so for example, an interactor may optionally see the main TV program image in a secondary PIP on the PC/PDA, while other viewers may optionally see that related interaction activity in a secondary PIP on the TV, thus viewing the main program without obstruction. The further dimension of how users can control the various Display Set Options is summarized in the section of Appendix A headed Control Options.

Focusing on the lower levels of system configuration and communications paths, the sections headed Direct signal sources and Signal relay list some of the alternative embodiments in cases where resource inputs may be directly sourced, or relayed from another device set.

Although the invention has been described in detail in the foregoing embodiments, it is to be understood that the descriptions have been provided for purposes of illustration only and that other variations both in form and detail can be made thereupon by those skilled in the art without departing from the spirit and scope of the invention, which is defined solely by the appended claims. Each of the above-noted references is incorporated herein by reference in its entirety.

Appendix A: Typical modes of coordination

Display Set Options

Interaction Element	Display set options----------	----------
	TV picture	PC/PDA screen
1. Main video program screen	• Main screen	• Duplicate main screen • Different main screen
2. Additional video PIP screen	• Standard PIP view • Mask out (omit) PIP	• Duplicate PIP screen • Only PIP screen • Reverse main/PIP
3. ITV bugs and cues	• Show on main screen • Show on PIP • Mask out	• Duplicate • Show here only • Show different options
4. ITV browser screen (or independent Web page) …or other "computer" screen such as e-mail, chat, etc.	• Show on main screen • Show on PIP • Mask out	• Duplicate • Show here only • Show different options o More options o Different people
5. Video window within browser screen	• Show on main screen • Show on PIP • Mask out	• Duplicate • Show here only • Show different options…
6. Interactive image hot-spots	• Show on main screen • Mask out	• Duplicate • Show here only • Show different options…

Control Options

Interaction Element	Control options-	----------
	TV remote control	PC/PDA controls
1-6 (any or all)	• Controls both devices • Controls TV only • Migrates ITV interactions to PC	• Controls both devices • Controls PC only • Migrates PC interactions (screen and state) back to TV

Direct signal sources

Interaction Element	Direct signal sources	----------
	TV picture	PC/PDA screen
1-6 (any or all)	• Broadcast • IP stream • Stored	• Broadcast • IP stream • Stored

Signal relay

Interaction Element	Signal relay-------	----------
	PC to TV	TV to PC
1-6 (any or all)	• Broadcast signal • A/V signal • IP stream • Via head-end	• Broadcast signal • A/V signal • IP stream • Via head-end

What is claimed is:

1. A method for operating a computer, comprising:

receiving a state record containing state information relating to a media player; and

establishing a session at said computer, said session based on said state information.

2. The method of claim 1, wherein said media player and said computer do not share operational logic.

3. The method of claim 1, wherein the media player is a digital video recorder.

4. The method of claim 1, wherein said media player is a television.

5. The method of claim 1, wherein said computer is a personal computer.

6. The method of claim 1, wherein said computer is a personal data assistant.

7. The method of claim 1, wherein the state record is used at the computer system to identify a hyperlink starting resource for said session.

8. The method of claim 1, wherein said media player and said computer use at least one of separate hardware logic instances and separate software process instances.

9. The method of claim 1, wherein a user may, while utilizing a session at said media player, employ a navigation control to select said computer.

10. The method of claim 9, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be the ending resource for a link activated at the first device set.

11. The method of claim 1, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be a resource currently presented at the first device set.

12. The method of claim 1, wherein said state record is accessible to said computer.

13. The method of claim 1, further comprising converting a resource corresponding to a session at said media player into a format suitable for presentation at said computer.

14. The method of claim 1, wherein said media player is one of a browser and an interactive television viewing system.

15. The method of claim 14, wherein the browser is a web browser.

16. The method of claim 1, wherein navigation actions at said computer set can target said media player.

17. The method of claim 1, wherein said media player comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, and a personal computer acting as a media gateway/server.

18. The method of claim 1, wherein said computer comprises one of a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

19. The method of claim 1, wherein a resource presented at said computer is the ending resource for a link activated at said media player.

20. The method of claim 19, wherein the media player is presenting a television-type program, and the ending resource comprises enhancement content associated with the television-type program.

21. The method of claim 20, wherein a user may choose to target some enhancement content resources to the media player, and other enhancement content resources to the computer.

22. The method of claim 19, wherein the ending resource comprises one of enhancement content associated with the television-type program, and an alternative television program resource and time position.

23. The method of claim 1, wherein an interactive television session is associated with said media player.

24. The method of claim 1, wherein a session relating to simple television viewing is presented at said media player, and the session associated with said computer relates to enhancement content associated with television viewing.

25. The method of claim 1, wherein the computer set is activated just prior to said establishing, without need for said media player to be specifically pre-configured to interact with said computer.

26. The method of claim 25, wherein one of a device set resource name and a device set network address is resolved dynamically.

27. The method of claim 1, wherein the computer is activated after said establishing without need for said media player to be specifically pre-configured to interact with said computer.

28. The method of claim 27, wherein one of a device set resource name and a device set network address is resolved dynamically.

29. The method of claim 1, wherein said state information comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

30. The method of claim 29, wherein said state information comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across said media player and said computer.

31. The method of claim 29, wherein a cookie is sent to a remote server from each of the media player and the computer in order to maintain a remote session with said server.

32. The method of claim 31, wherein the remote session employs one of HTTP and HTTPS.

33. The method of claim 31, wherein, through the use of a proxy server, said remote server views said media player and said computer as having identical network identities.

34. The method of claim 33, wherein the actual network identity corresponding to said media player is not identical to the actual network identity corresponding to said computer.

35. The method of claim 33, wherein said network identities comprise network addresses.

36. The method of claim 33, wherein said proxy server is collocated with said media player.

37. The method of claim 1, wherein said state record comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

38. The method of claim 37, wherein said state record comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across said media player and said computer.

39. The method of claim 37, wherein a cookie is sent to a remote server from each of the media player and the computer in order to maintain a remote session with said server.

40. The method of claim 39, wherein remote session employs one of HTTP and HTTPS.

41. The method of claim 39, wherein, through the use of a proxy server, said remote server views said media player and said computer as having identical network identities.

42. The method of claim 41, wherein the actual network identity corresponding to said media player is not identical to the actual network identity corresponding to said computer.

43. The method of claim 41, wherein said network identities comprise network addresses.

44. The method of claim 41, wherein said proxy server is collocated with said media player.

45. The method of claim 1, wherein at least one of said computer and said media player has a form factor suited to across-the-room viewing and readily viewed by multiple users, and at least one of said computer and said media player has a form factor suited to close work by a single user.

46. The method of claim 1, further comprising:

additionally establishing a session at a device set, said session based on said state information,

wherein said computer and said device set are allocated for use by different users.

47. The method of claim 46, wherein said computer and said device set are used independently.

48. The method of claim 46, wherein said different users perform navigation actions that target the media player.

49. The method of claim 1, wherein rules limit the ability of a user to target a multi-user device set.

50. The method of claim 1, wherein multiuser collaboration features are supported.

51. The method of claim 1, wherein said media player and said computer are coordinated by a central controller.

52. The method of claim 1, wherein said media player and said computer sets are coordinated on a peer-to-peer basis without a central controller.

53. The method of claim 1, wherein a remote session state, associated with a remote system in communication with the media player, is further associated with said computer.

54. The method of claim 53, wherein said media player and said computer have different network addresses.

55. The method of claim 54, wherein said network addresses comprise IP addresses.

56. The method of claim 1, wherein said establishing occurs during an intermediate step in a remote session comprising the processing of a transaction with a remote server, wherein the continuity of said remote session is maintained.

57. The method of claim 1, wherein a user may set preferences that specify a default targeting of links to device sets.

58. The method of claim 57, wherein the user preferred default target is selectively specified in terms of at least one of: types of starting resources, types of ending resources, metadata attributes of starting resources, metadata attributes of ending resources, attributes of links, application categories, and defined alternative navigation modes.

59. The method of claim 1, wherein a resource presented at said media player may contain recommended targeting behavior attributes that specify a default preferred targeting of links to classes of device sets.

60. The method of claim 57, wherein said default targeting is specified in terms of at least one of defined alternative navigation modes, and the active states of the device sets.

61. The method of claim 1, wherein a link to an ending resource is specified, external to a starting resource presented to said media player, as a third party arc.

62. The method of claim 1, wherein a starting resource presented to said media player is one of 1) a resource containing links as outbound arcs, 2) a resource associated with external third-party arcs, 3) a resource containing coding other than links that specifies associations to other resources that give the effect of links, and 3) a resource associated with external coding that specifies associations to other resources that give the effect of links.

63. The method of claim 62, wherein said starting resource is distributed with associated third party arcs.

64. The method of claim 62, wherein an independent association is made with third-party arcs that are sourced separately from the starting resource.

65. The method of claim 62, wherein said third party arcs are associated with an entire linear resource.

66. The method of claim 62, wherein said third party arcs are associated with a time span for a continuous media resource.

67. The method of claim 63, wherein said third party arcs are associated with a region of an image resource.

68. The method of claim 62, wherein said coding other than links includes at least one of a menu structure having a defined graph structure, a transaction request form having a defined address for transaction submission, and a selectable content element having a defined address for related action.

69. The method of claim 1, wherein said establishing comprises targeting a defined display subunit of said computer, said display subunit comprising one of a screen area, a pane, a frame, a window, a picture-in-picture region, a physical subunit, a virtual subunit, and a relevant presentation context.

70. The method of claim 69, wherein said establishing may further comprise specification of attributes to select among full-screen, a specified intermediate size and position, a previous intermediate size and position, and a minimized display on said computer.

71. The method of claim 1, wherein a resource presented is obtained from at least one of a persistent storage device and a communications channel.

72. The method of claim 71, wherein said persistent storage device and said communications channel are connected via one of a) one of said media player and said computer, and b) each of said media player and said computer.

73. The method of claim 1, wherein coordination between said media player and said computer is effected using communications paths that link one of a) said media player and said computer directly to one another, b) said media player and said computer via an intermediate gateway system serving both, c) one of said media player and said computer via an intermediate gateway system serving the one of said media player and said computer, d) a local control system serving one of said media player and said computer , e) a local control system serving both said media player and said computer, f) a remote control system serving one of said media player and said computer, and g) a remote control system serving both of said media player and said computer.

74. The method of claim 1, wherein said media player and said computer share network access facilities.

75. The method of claim 1, wherein said media player and said computer are independently provisioned.

76. The method of claim 1, wherein at least one of internet protocol, digital television, Internet, cable, satellite, broadcast, mobile, and digital versatile disk is employed.

77. The method of claim 1, wherein resources and links associated with said session are defined by at least one of hypertext markup language, XML, extensible markup language linking language, motion picture experts group specifications, advanced television enhancement forum specifications, and synchronized multimedia integration language.

78. The method of claim 1, wherein said computer is selected based upon on one of a) attributes ordinarily used to control the targeting of navigation using a single display device, and b) attributes specific to presentation using multiple device sets.

79. The method of claim 1, wherein said second device set is selected based upon attributes comprising one of a) a new presentation context, b) a replacement presentation context that loads the presentation of an ending resource in the same context in which the presentation of a starting resource was loaded, c) an embedded presentation context that loads the presentation of an ending resource in place of a starting resource.

80. The method of claim 1, wherein said establishing comprises employing as a presentation context one or more of a screen area, a pane, a frame, a window, and a picture-in-picture region.

81. The method of claim 78, wherein said attributes ordinarily used to control the targeting of navigation using a single display device are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

82. The method of claim 78, wherein said attributes specific to presentation using multiple device sets are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

83. The method of claim 81, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

84. The method of claim 82, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

85. The method of claim 81, wherein said hypertext markup language attributes are hypertext markup language target attributes.

86. The method of claim 82, wherein said hypertext markup language attributes are hypertext markup language target attributes.

87. The method of claim 81, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

88. The method of claim 82, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

89. The method of claim 81, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

90. The method of claim 82, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

91. The method of claim 1, wherein said session is presented in accord with one of extensible markup language linking language actuate attributes hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes, and extensible hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes.

92. The method of claim 1, wherein said computer is selected such that there is no change in selection preference settings that affect subsequent link navigation steps unless a user

performing the selection indicates that there be a change in selection preference settings that affect subsequent link navigation steps.

93. The method of claim 1, wherein a user preferred default selection of said computer is pre-specified by a third party, including a system component provider.

94. The method of claim 1, wherein said establishing is triggered by a user.

95. The method of claim 1, wherein said establishing is triggered at any of multiple stages in a session associated with said media player.

96. The method of claim 1, wherein said computer is selected from computers of varying form factors, wherein resources associated with a session associated with said media player are altered only with regard to presentation adaptations needed to adapt said resources to the form factor of said computer.

97. The method of claim 96, wherein said adaptations relate to at least one of display parameters and input device parameters.

98. The method of claim 97, wherein said display parameters comprise at least one of size, resolution, and color.

99. The method of claim 97, wherein said input device parameters comprise at least one of keyboard, stylus, touch screen, mouse, and buttons.

100. The method of claim 1, wherein said establishing maintains user state at said media player.

101. The method of claim 1, wherein said establishing maintains server state for a server that was allied, at the time of said establishing, with a session associated with said media player. at the time of continuing.

102. The method of claim 1, wherein state continuation after said establishing is supported at a granularity corresponding to completed user interaction steps and user-visible results.

103. The method of claim 102, wherein said state continuation does not maintain details of internal program execution state of software processes that respond to a user's interaction but are not visible to the user.

104. The method of claim 1, wherein state continuation after said establishing is supported at a granularity corresponding to the identity, time-position and presentation state of a browsed resource.

105. The method of claim 104, wherein said state continuation need not maintain details of user interaction steps regarding data entry inputs to be sent to a resource.

106. The method of claim 1, wherein state continuation after said establishing is supported at a granularity corresponding to the identity of a browsed resource.

107. The method of claim 106, wherein said state continuation need not maintain details of time position and user interaction steps regarding a resource.

108. The method of claim 1, wherein an interactive television session is presented at said media player, and said computer is at one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

109. The method of claim 1, where said media player is a lean-back configuration, and said computer is a lean-forward configuration.

110. The method of claim 1, wherein said establishing is triggered at a loading of a content resource.

111. The method of claim 1, wherein said establishing is triggered at a traversal of a link.

112. The method of claim 1, wherein said establishing is triggered at an intermediate state in the play of a continuous media content resource.

113. The method of claim 1, wherein said establishing is triggered when a user interaction with the presentation of a content resource is complete.

114. The method of claim 113, wherein said content resource is a form.

115. The method of claim 114, wherein said user interaction is the filling out of said form.

116. The method of claim 1, wherein resources accessed in said session remain independent of said establishing.

117. The method of claim 96, wherein said adaptations are accommodated by adaptations performed at said computer after said computer accesses said resources.

118. The method of claim 96, wherein said adaptations are accommodated by adaptations performed, at a remote server, after an access to said resources executed by said computer.

119. The method of claim 96, wherein no intermediary server is used to adapt said resources.

120. The method of claim 117, wherein said accommodation is guided by coding, associated with said resources, that specifies recommended accommodations corresponding to varying form factors.

121. The method of claim 1, further comprising moving said session from said computer back to said media player.

122. The method of claim 1, wherein said establishing is independent of further actions at said media player.

123. The method of claim 122, wherein a session relating to said media player is forked into a first session corresponding to said media player and a second session corresponding to said computer, the two sessions utilizing two separate browser instances.

124. The method of claim 123, wherein the separate browser instance corresponding to one of said media player and said computer is not affected by further browsing actions at the other of said media player and said computer unless requested by a user.

125. The method of claim 1, wherein said session is additional to a session already established at said computer.

126. The method of claim 1, wherein said session replaces a specified session at said computer.

127. The method of claim 122, wherein a targeting of one of said media player and said computer from the other of said media player and said computer is designated to trigger a further establishment.

128. The method of claim 127, wherein the further establishment is designated to do one of 1) creating a new session at the targeted one of said media player and said computer that is additional to an existing session there, and 2) replacing the current state of a specified session at the targeted one of said media player and said computer.

129. The method of claim 1, wherein said establishing involves one of 1) moving all browser instances associated with a session allied with said media player, and 2) moving only the browser instance from which said establishing is initiated.

130. The method of claim 1, wherein said establishing is triggered by activating said computer.

131. The method of claim 1, wherein said establishing is triggered by deactivating said media player.

132. The method of claim 1, Wherein said state information is communicated from a browser instance at media player set to a browser instance at said computer set at the time of establishing.

133. The method of claim 132, wherein state information changing events prior to the establishment are not individually logged.

134. The method of claim 132, wherein, and said state record is single state record that defines the current state information relating to said media player at said time.

135. The method of claim 1, wherein a simple event history, in the form of a history list that records the identity and sequence of resources navigated and a current list of the recent sequence of that resource navigation, is provided in said state record as a current state record element.

136. The method of claim 1, wherein the method is employable by browser processes differing by one of different software code and different hardware architectures.

137. The method of claim 136, wherein said browser processes are capable of exporting and importing said state record using a defined common format.

138. The method of claim 137, wherein the method and the defined common format is made available for use by any provider of browsers under reasonable and non-discriminatory terms.

139. The method of claim 1, wherein said establishing is triggered by a user transfer request signal.

140. The method of claim 1, wherein said establishing is triggered by one of navigation to a triggering resource and navigation of a triggering link.

141. The method of claim 140, wherein the triggering resource is a resource belonging to a category pre-defined to cause triggering.

142. The method of claim 140, wherein the triggering resource is a resource containing a specific attribute specifying a trigger.

143. The method of claim 140, wherein the category pre-defined to cause triggering is one of a navigation level, a content type, and an activity type.

144. The method of claim 133, wherein maintenance of information needed to transfer cumulative state is routinely performed.

145. The method of claim 144, wherein said maintenance is routinely performed once support for said media player and said computer is first enabled at one of system setup time and device set setup time.

146. The method of claim 144, wherein said maintenance does not depend on a user activating a special state tracking activity for said session.

147. The method of claim 1, further comprising:

said media player providing user identification information corresponding to a user; and

said computer providing user identification information corresponding to said user.

148. The method of claim 147, wherein user authentication information is provided by both said media player and said computer.

149. The method of claim 1, wherein said state information is used at said computer to identify a resource for browsing.

150. The method of claim 149, wherein the resource is used as a hyperlink starting resource with one of a designated set of third-party arcs to determine a hyperlink ending resource for traversal at said computer.

151. The method of claim 149, wherein said state record comprises a current time position and a current time.

152. The method of claim 149, wherein said establishing is triggered by a user signal.

153. The method of claim 149, wherein establishing is performed when a program change event occurs.

154. The method of claim 150, wherein the hyperlink ending resource is presented to a user for activation.

155. The method of claim 1, further comprising:

transmitting said state record from said media player to said computer, wherein said media player set initiates the transmission.

156. The method of claim 1, further comprising:

said computer requesting said state record.

157. The method of claim 1, further comprising:

acquiring information of the identity of a resource associated with a session presented at said media player; and

employing said information to determine a source for enhancement resources related to said session presented at said media player.

158. The method of claim 157, wherein the information is obtained via a wide area network.

159. The method of claim 158, wherein the wide area network is the Internet.

160. The method of claim 1, wherein the state record includes a navigation history, wherein said navigation history can be retraced as if said continuing had not occurred.

161. A method of providing a browsing session comprising:

beginning a session comprising browsing interactions at a first device set; and

continuing said session at a second device set.

162. The method of claim 161, wherein a resource presented at the second device set is the ending resource for a link activated at the first device set.

163. The method of claim 161, wherein said first device set and said second device set use different browser instances.

164. The method of claim 161, wherein said first device set and said second device set use at least one of separate hardware logic instances and separate software process instances.

165. The method of claim 161, wherein a user may, while utilizing said session at said first device set, employ a navigation control to select said second device set.

166. The method of claim 165, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be the ending resource for a link activated at the first device set.

167. The method of claim 165, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be a resource currently presented at the first device set.

168. The method of claim 161, wherein state data corresponding to said session at said first device set is provided to said second device set.

169. The method of claim 161, wherein state data corresponding to said session at said first device set is accessible to said second device set.

170. The method of claim 161, further comprising converting a resource corresponding to said session at said first device into a format suitable for presentation at said second device set.

171. The method of claim 161, wherein said second device is one of a browser and an interactive television viewing system.

172. The method of claim 171, wherein the browser is a web browser.

173. The method of claim 161, wherein navigation actions at the second device set can target the first device set.

174. The method of claim 161, wherein said first device set comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

175. The method of claim 161, wherein said second device set comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

176. The method of claim 161, wherein said first device set and said second device set are of dissimilar types.

177. The method of claim 162, wherein the first device set is a television-type device set presenting a television-type program, the second device set is a computer-type device set, and the ending resource comprises enhancement content associated with the television-type program.

178. The method of claim 177, wherein a user may choose to target some enhancement content resources to the television-type device, and other enhancement content resources to the personal computer-type device.

179. The method of claim 162, wherein the second device set is a television-type device set presenting at least one of a television-type program and enhancements, the first device set is a computer-type device set, and the ending resource comprises one of enhancement content associated with the television-type program, and an alternative television program resource and time position.

180. The method of claim 161, wherein said session comprises an interactive television session, the first device set is a television, and the second device set is one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

181. The method of claim 161, wherein the first device set is a lean-back configuration, and the second device set is a lean-forward configuration.

182. The method of claim 161, wherein said session as presented at said first device set relates to simple television viewing, and said session as presented at said second device set relates to enhancement content associated with television viewing.

183. The method of claim 161, wherein the second device set is activated just prior to said continuing, without need for said first device set to be specifically pre-configured to interact with said second device set.

184. The method of claim 183, wherein one of a device set resource name and a device set network address is resolved dynamically.

185. The method of claim 161, wherein the second device set is activated after said continuing without need for said first device set to be specifically pre-configured to interact with said second device set.

186. The method of claim 185, wherein one of a device set resource name and a device set network address is resolved dynamically.

187. The method of claim 186, wherein said state data comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

188. The method of claim 187, wherein said state data comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across the two device sets.

189. The method of claim 187, wherein a cookie is sent to a remote server from each of the first and second device sets in order to maintain a remote session with said server.

190. The method of claim 189, wherein the remote session employs one of HTTP and HTTPS.

191. The method of claim 189, wherein, through the use of a proxy server, said remote server views said first device set and said second device set as having identical network identities.

192. The method of claim 191, wherein the actual network identity corresponding to said first device set is not identical to the actual network identity corresponding to said second device set.

193. The method of claim 191, wherein said network identities comprise network addresses.

194. The method of claim 191, wherein said proxy server is collocated with said first device set.

195. The method of claim 169, wherein said state data comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

196. The method of claim 195, wherein said state data comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across the two device sets.

197. The method of claim 195, wherein a cookie is sent to a remote server from each of the first and second device sets in order to maintain a remote session with said server.

198. The method of claim 197, wherein remote session employs one of HTTP and HTTPS.

199. The method of claim 197, wherein, through the use of a proxy server, said remote server views said first device set and said second device set as having identical network identities.

200. The method of claim 199, wherein the actual network identity corresponding to said first device set is not identical to the actual network identity corresponding to said second device set.

201. The method of claim 199, wherein said network identities comprise network addresses.

202. The method of claim 199, wherein said proxy server is collocated with said first device set.

203. The method of claim 161, wherein at least one of said device sets has a form factor suited to across-the-room viewing and readily viewed by multiple users, and at least one of said device sets has a form factor suited to close work by a single user.

204. The method of claim 161, further comprising:

additionally continuing said session at a third device set,

wherein said second device set and said third device set are allocated for use by different users.

205. The method of claim 204, wherein said second device set and said third device set are used independently.

206. The method of claim 204, wherein said different users perform navigation actions that target the first device set.

207. The method of claim 161, wherein rules limit the ability of a user to target a multi-user device set.

208. The method of claim 161, wherein multiuser collaboration features are supported

209. The method of claim 161, wherein said device sets are coordinated by a central controller.

210. The method of claim 161 wherein said device sets are coordinated on a peer-to-peer basis without a central controller.

211. The method of claim 161, wherein a remote session state, associated with a remote system in communication with the first device set, is further associated with said second device set.

212. The method of claim 211, wherein the first and second device sets have different network addresses.

213. The method of claim 212, wherein said network addresses comprise IP addresses.

214. The method of claim 161, wherein said continuing occurs during an intermediate step in a remote session comprising the processing of a transaction with a remote server, wherein the continuity of said remote session is maintained.

215. The method of claim 161, wherein a user may set preferences that specify a default targeting of links to device sets.

216. The method of claim 215, wherein the user preferred default target is selectively specified in terms of at least one of: types of starting resources, types of ending resources, metadata attributes of starting resources, metadata attributes of ending resources, attributes of links, application categories, and defined alternative navigation modes.

217. The method of claim 161, wherein a resource presented at said first device set may contain recommended targeting behavior attributes that specify a default preferred targeting of links to classes of device sets.

218. The method of claim 215, wherein said default targeting is specified in terms of at least one of defined alternative navigation modes, and the active states of the device sets.

219. The method of claim 161, wherein a link to an ending resource is specified, external to a starting resource presented to said first device set, as a third party arc.

220. The method of claim 161, wherein a starting resource presented to said first device set is one of 1) a resource containing links as outbound arcs, 2) a resource associated with external third-party arcs, 3) a resource containing coding other than links that specifies associations to other resources that give the effect of links, and 3) a resource associated with external coding that specifies associations to other resources that give the effect of links.

221. The method of claim 220, wherein said starting resource is distributed with associated third party arcs.

222. The method of claim 220, wherein an independent association is made with third-party arcs that are sourced separately from the starting resource.

223. The method of claim 220, wherein said third party arcs are associated with an entire linear resource.

224. The method of claim 221, wherein said third party arcs are associated with a time span for a continuous media resource.

225. The method of claim 221, wherein said third party arcs are associated with a region of an image resource.

226. The method of claim 220, wherein said coding other than links includes at least one of a menu structure having a defined graph structure, a transaction request form having a defined address for transaction submission, and a selectable content element having a defined address for related action.

227. The method of claim 161, wherein said continuing comprises targeting a defined display subunit of said second device set, said display subunit comprising one of a screen area, a pane, a frame, a window, a picture-in-picture region, a physical subunit, a virtual subunit, and a relevant presentation context.

228. The method of claim 227, wherein said continuing may further comprise specification of attributes to select among full-screen, a specified intermediate size and position, a previous intermediate size and position, and a minimized display on said second device set.

229. The method of claim 161, wherein a resource presented in conjunction with said session is obtained from at least one of a persistent storage device and a communications channel.

230. The method of claim 229, wherein said persistent storage device and said communications channel are connected via one of a) one of the device sets, and b) each of the devices sets.

231. The method of claim 161, wherein coordination between the device sets is effected using communications paths that link one of a) the device sets directly to one another, b) the device sets via an intermediate gateway system serving both, c) one of the device sets via an intermediate gateway system serving that device set, d) a local control system serving one of said device sets, e) a local control system serving both of said device sets, f) a remote control system serving one of said device sets, and g) a remote control system serving both of said device sets.

232. The method of claim 161, wherein each of said device sets has the capability of operating as a browser entirely independently of the other of said device sets.

233. The method of claim 232, wherein said device sets share network access facilities.

234. The method of claim 232, wherein the device sets are independently provisioned.

235. The method of claim 161, wherein at least one of internet protocol, digital television, Internet, cable, satellite, broadcast, mobile, and digital versatile disk is employed.

236. The method of claim 161, wherein resources and links associated with said session are defined by at least one of hypertext markup language, extensible markup language,

extensible markup language linking language, Motion Picture Experts Group specifications, advanced television enhancement forum specifications, and synchronized multimedia integration language.

237. The method of claim 161, wherein said second device set is selected based upon on one of a) attributes ordinarily used to control the targeting of navigation using a single display device, and b) attributes specific to presentation using multiple device sets.

238. The method of claim 161, wherein said second device set is selected based upon attributes comprising one of a) a new presentation context, b) a replacement presentation context that loads the presentation of an ending resource in the same context in which the presentation of a starting resource was loaded, c) an embedded presentation context that loads the presentation of an ending resource in place of a starting resource.

239. The method of claim 161, wherein said continuing comprises employing as a presentation context one or more of a screen area, a pane, a frame, a window, and a picture-in-picture region.

240. The method of claim 237, wherein said attributes ordinarily used to control the targeting of navigation using a single display device are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

241. The method of claim 237, wherein said attributes specific to presentation using multiple device sets are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

242. The method of claim 240, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

243. The method of claim 241, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

244. The method of claim 240, wherein said hypertext markup language attributes are hypertext markup language target attributes.

245. The method of claim 241, wherein said hypertext markup language attributes are hypertext markup language target attributes.

246. The method of claim 240, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

247. The method of claim 241, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

248. The method of claim 240, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

249. The method of claim 241, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

250. The method of claim 161, wherein said session is presented in accord with one of extensible markup language linking language actuate attributes hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes, and extensible hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes.

251. The method of claim 161, wherein said second device set is selected such that there is no change in selection preference settings that affect subsequent link navigation steps unless a user performing the selection indicates that there be a change in selection preference settings that affect subsequent link navigation steps.

252. The method of claim 161, wherein said second device set is selected such that said second device set is assumed to be the first device set unless indicated otherwise by one of user action at the time of selection, user preferences established prior to the time of selection, and content author recommendations.

253. The method of claim 161, wherein a user preferred default selection of said second device set is pre-specified by a third party, including a system component provider.

254. The method of claim 161, wherein said continuing is triggered by a user.

255. The method of claim 161, wherein said continuing is triggered at any of multiple stages in said session.

256. The method of claim 161, wherein said second device set is selected from device sets of varying form factors, wherein resources associated with said session are altered only with regard to presentation adaptations needed to adapt said resources to the form factor of said second device set.

257. The method of claim 256, wherein said adaptations relate to at least one of display parameters and input device parameters.

258. The method of claim 257, wherein said display parameters comprise at least one of size, resolution, and color.

259. The method of claim 257, wherein said input device parameters comprise at least one of keyboard, stylus, touch screen, mouse, and buttons.

260. The method of claim 161, wherein said continuing maintains user state at said first device set.

261. The method of claim 161, wherein said continuing maintains server state for a server that was associated with said session at the time of continuing.

262. The method of claim 161, wherein state continuation after said continuing is supported at a granularity corresponding to completed user interaction steps and user-visible results.

263. The method of claim 262, wherein said state continuation does not maintain details of internal program execution state of software processes that respond to a user's interaction but are not visible to the user.

264. The method of claim 161, wherein state continuation after said continuing is supported at a granularity corresponding to the identity, time-position and presentation state of a browsed resource.

265. The method of claim 264, wherein said state continuation need not maintain details of user interaction steps regarding data entry inputs to be sent to a resource.

266. The method of claim 161, wherein state continuation after said continuing is supported at a granularity corresponding to the identity of a browsed resource.

267. The method of claim 266, wherein said state continuation need not maintain details of time position and user interaction steps regarding a resource.

268. The method of claim 161, wherein the session comprises an interactive television session, and said first device is a television and said second device is at one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

269. The method of claim 161, wherein said continuing is triggered at a loading of a content resource.

270. The method of claim 161, wherein said continuing is triggered at a traversal of a link.

271. The method of claim 161, wherein said continuing is triggered at an intermediate state in the play of a continuous media content resource.

272. The method of claim 161, wherein said continuing is triggered when a user interaction with the presentation of a content resource is complete.

273. The method of claim 272, wherein, said content resource is a form.

274. The method of claim 273, wherein said user interaction is the filling out of said form.

275. The method of claim 161, wherein resources accessed in said session remain independent of said continuing.

276. The method of claim 256, wherein said adaptations are accommodated by adaptations performed at said second device set after said second device set accesses said resources.

277. The method of claim 256, wherein said adaptations are accommodated by adaptations performed, at a remote server, after an access to said resources executed by said second device set.

278. The method of claim 256, wherein no intermediary server is used to adapt said resources.

279. The method of claim 276, wherein said accommodation is guided by coding, associated with said resources, that specifies recommended accommodations corresponding to varying form factors.

280. The method of claim 161, further comprising moving said session from said second device set back to said first device set.

281. The method of claim 161, wherein said continuing is independent of further actions at said first device set.

282. The method of claim 281, wherein said session is forked into a first session corresponding to said first device set and a second session corresponding to said second device set, the two sessions utilizing two separate browser instances.

283. The method of claim 282, wherein the separate browser instance corresponding to one of said device sets is not affected by further browsing actions at the other of said device sets unless requested by a user.

284. The method of claim 281, wherein said continuing is designated to do one of 1) creating a new session at the second device set that is additional to an existing session there, and 2) replacing the current state of a specified session at the second device set.

285. The method of claim 281, wherein a targeting of one of said device sets from the other of said device sets is designated to trigger a further continuance.

286. The method of claim 285, wherein the further continuance is designated to do one of 1) creating a new session at the targeted device set that is additional to an existing session there, and 2) replacing the current state of a specified session at said targeted device set.

287. The method of claim 161, wherein said continuing involves one of 1) moving all browser instances associated said session as presented at said first device set, and 2) moving only the browser instance from which said continuing is initiated.

288. The method of claim 161, wherein said continuing is triggered by activating the second device set.

289. The method of claim 161, wherein said continuing is triggered by deactivating the first device set.

290. The method of claim 161, Wherein the state of the session is communicated from a browser instance at the first device set to a browser instance at the second device set at the time of continuing.

291. The method of claim 290, wherein session state change events prior to the continuation are not individually logged.

292. The method of claim 290, wherein the state of the session is transferred as a single state record that defines the current cumulative state of the session at said time.

293. The method of claim 291, wherein a simple event history, in the form of a history list that records the identity and sequence of resources navigated and a current list of the recent

sequence of that resource navigation, is provided in the state record as a current state record element.

294. The method of claim 161, wherein the method is employable by browser processes differing by one of different software code and different hardware architectures.

295. The method of claim 294, wherein said browser processes are capable of exporting and importing a state record corresponding to said session using a defined common format.

296. The method of claim 295, wherein the method and the defined common format is made available for use by any provider of browsers under reasonable and non-discriminatory terms.

297. The method of claim 161, wherein said continuing is triggered by a user transfer request signal.

298. The method of claim 161, wherein said continuing is triggered by one of navigation to a triggering resource and navigation of a triggering link.

299. The method of claim 298, wherein the triggering resource is a resource belonging to a category pre-defined to cause triggering.

300. The method of claim 298, wherein the triggering resource is a resource containing a specific attribute specifying a trigger.

301. The method of claim 298, wherein the category pre-defined to cause triggering is one of a navigation level, a content type, and an activity type.

302. The method of claim 291, wherein maintenance of information needed to transfer cumulative state is routinely performed.

303. The method of claim 302, wherein said maintenance is routinely performed once support for the said device sets is first enabled at one of system setup time and device set setup time.

304. The method of claim 302, wherein said maintenance does not depend on a user activating a special state tracking activity for said session.

305. The method of claim 161, further comprising:

said first device set providing user identification information corresponding to a user; and

said second device set providing user identification information corresponding to said user.

306. The method of claim 305, wherein user authentication information is provided by both device sets.

307. The method of claim 161, further comprising:

transmitting from said first device set to said second device set a state record corresponding to said session as presented at said first device set.

308. The method of claim 307, wherein the state record is used at said second device set to identify a resource for browsing.

309. The method of claim 308, wherein the resource is used as a hyperlink starting resource with one of a designated set of third-party arcs to determine a hyperlink ending resource for traversal at said second device set.

310. The method of claim 307, wherein the state record comprises a current time position and a current time.

311. The method of claim 307, wherein said transmitting is triggered by a user signal.

312. The method of claim 307, wherein said transmitting is performed when a program change event occurs.

313. The method of claim 309, wherein the hyperlink ending resource is presented to a user for activation.

314. The method of claim 307, wherein said first device set initiates the transmission of said state record.

315. The method of claim 307, wherein said state record is requested from said first device set by said second device set.

316. The method of claim 161, further comprising:

acquiring information of the identity of a resource associated with said session as presented at said first device set; and

employing said information to determine a source for enhancement resources related to said session as presented at said first device set.

317. The method of claim 316, wherein the information is obtained via a wide area network.

318. The method of claim 317, the wide area network is the Internet.

319. The method of claim 307, wherein the state record includes a navigation history, wherein said navigation history can be retraced as if said continuing had not occurred.

320. A method of presenting hypermedia, comprising:

presenting to a user at a first device set a starting resource, said starting resource having associated links to an ending resource; and

presenting said ending resource at a selected device set, wherein said device sets use separate presentation process instances.

321. The method of claim 320, wherein the selection is specific to the link being traversed.

322. The method of claim 320, wherein the selected device set is chosen by the user.

323. The method of claim 320, wherein the selected device set is indicated by the creator of said starting resource.

324. The method of claim 320, further comprising the step of modifying said ending resource for suitable presentation at said selected device set.

325. A method of conducting a browsing session, comprising:

beginning a session comprising browsing interactions at a first device set; and

continuing said session at a second device set, wherein the change in device set does not substantively alter the state of the session.

326. The method of claim 325, wherein said continuing is in response to a trigger.

327. The method of claim 326, wherein said trigger is provided by a user effecting said interactions.

328. The method of claim 326, wherein said trigger is provided by a content resource used in said session.

329. The method of claim 325, further comprising the step of modifying said state of the session to suit said second device set.

330. The method of claim 325, wherein said device sets use separate browser process instances.

331. The method of claim 325, wherein said device sets utilize separately stored session state information.

332. A method for associating remote sessions with user systems, comprising:

conducting a remote session in association with a first user device set;

associating with said first user device set an identifier corresponding to a user associated with said remote session;

receiving from a second user device set a communication including said identifier; and

continuing said remote session with said second user device set.

333. The method of claim 332, wherein said session comprises browsing interactions.

334. The method of claim 332, wherein user authentication information is provided by both device sets.

335. The method of claim 332, wherein said identifier comprises a cookie.

336. A method for continuing a remote session, comprising:

conducting a remote session from a first user device set;

transmitting from a second user device set a communication including an identifier corresponding to the user associated with said second user device set; and

continuing said remote session at said second user device set.

337. A method of conducting a browsing session, comprising:

beginning a session comprising browsing interactions at a first device set, there being a first user identity at said first device set;

having a remote server relay from said first device set to a second device set data relating to said session, there being is a second user identity at said second device set; and

continuing said session at said second device set, wherein the change in device set does not substantively alter the state of the session,

wherein said first user identity corresponds to the same user as said second user identity.

338. A method of operating a television system head-end, comprising:

receiving state information from a set-top box; and

relaying said state information over an Internet connection to a device set collocated with said set-top box.

339. The method of claim 338, wherein a subscription fee is charged for the relaying service.

340. A method of operating a television system head-end, comprising:

receiving remote control commands over an Internet connection; and

relaying said remote control commands to a designated set-top box.

341. A method for relaying sessions to remote client systems, comprising:

receiving from a first user device set a session state record corresponding to a session at said first user device set;

associating with said first user device set an identifier corresponding to a user;

receiving from a second user device set a communication including said identifier; and

transferring at least a portion of said state record to said second user device set.

342. The method of claim 341, wherein said session comprises browsing interactions.

343. A method for continuing a session, comprising:

transmitting from a first user device set a communication including an identifier corresponding to a user; and

receiving at said first user device set a session state record corresponding to a session previously begun at a second user device set.

344. A method of operating a television system, comprising:

obtaining a record of a user's interaction with a television system; and

relaying said record to an independent computer at the user's location to facilitate further interactions related to said interaction with a television system.

345. The method of claim 344, wherein said record is a message containing a link to a web-based service.

346. The method of claim 344, wherein said record is a transfer state record that is used at the computer to establish a related interactive session.

347. The method of claim 344, wherein said further interactions are facilitated using information obtained from said computer.

348. The method of claim 344, wherein said relaying comprises using a head-end server.

349. The method of claim 344, wherein said relaying comprises using interactive television browsing software running on a set top box.

350. The method of claim 344, wherein said relaying comprises using special purpose relay software running on a set top box.

351. A method of conducting a browsing session comprising:

beginning a session comprising browsing interactions;

interrupting said session;

storing a record of the state of said session; and

employing said record to continue said session from the point of interruption.

352. The method of claim 351, wherein the record of the state includes a navigation history.

353. The method of claim 352, wherein said navigation history can be retraced as if said continuing had not occurred.

354. The method of claim 351, wherein the record of the state is stored at the location of a user device set.

355. The method of claim 351, wherein said session is continued at a device set other than the device set at which is was interrupted.

356. A method of operating a media player, comprising:

interposing a device between a media player and a remote control device;

employing the interposed device to obtain state information relating to the media player; and

establishing a session at a computer, said session being based on said state information.

357. The method of claim 356, wherein the interposed device is a digital video recorder.

358. The method of claim 356, wherein the interposed device is a liberator device.

359. The method of claim 356, wherein said media player is a television.

360. The method of claim 356, wherein said computer is a personal computer.

361. The method of claim 356, wherein said computer is a personal data assistant.

362. The method of claim 356, wherein the state information is used at the computer system to identify a hyperlink starting resource for said session.

363. A method for receiving hyperlinks for interactive television, comprising:

submitting to a linkbase provider a request for one or more hyperlinks, said request including interactive television state information corresponding to a particular device; and

receiving from said provider a response to said request, the provider's formulation of said response comprising consulting a linkbase correlating state information with hyperlinks,

wherein requests relating to any of multiple programs are submitted, said requests differing only with regard to said state information.

364. The method of claim 363, wherein said request is dispatched via a computer network.

365. The method of claim 365, wherein said network is the internet.

366. The method of claim 363, wherein said linkbase is an array.

367. The method of claim 363, wherein said linkbase is a database.

368. The method of claim 363, wherein said linkbase is a process.

369. The method of claim 363, wherein the interactive television state information corresponding to a particular device includes an indication of channel.

370. The method of claim 363, wherein the interactive television state information corresponding to a particular device includes an indication of broadcast time.

371. The method of claim 363, wherein the correlated state information includes indications of channels.

372. The method of claim 363, wherein the correlated state information includes indications of broadcast times.

373. A method for providing hyperlinks for interactive television, comprising:

receiving hyperlink data associated with a television program; and

employing said hyperlink data in the creation of a linkbase, said linkbase correlating state information with hyperlinks.

374. The method of claim 373, wherein said linkbase is stored for future use.

375. The method of claim 373, wherein the method is employed by a television service operator to offer a common portal service that supports multiple programs offered by that operator.

376. The method of claim 373, wherein said linkbase is an array.

377. The method of claim 373, wherein said linkbase is a database.

378. The method of claim 373, wherein said linkbase operates as a process.

379. The method of claim 373, wherein the interactive television state information corresponding to a particular device includes an indication of channel.

380. The method of claim 373, wherein the interactive television state information corresponding to a particular device includes an indication of broadcast time.

381. The method of claim 373, wherein the correlated state information includes indications of channels.

382. The method of claim 373, wherein the correlated state information includes indications of broadcast times.

383. A method for providing hyperlinks for interactive television, comprising:

having access to a linkbase, said linkbase correlating state information with hyperlinks;

receiving a request for one or more hyperlinks, said request including interactive television state information corresponding to a particular device; and

employing said linkbase to respond to said request, wherein requests relating to any of multiple programs are submitted, said requests differing only with regard to said state information.

384. The method of claim 383, wherein the method is employed by a television service operator to offer a common portal service that supports multiple programs offered by that operator.

385. The method of claim 383, wherein said linkbase is an array.

386. The method of claim 383, wherein said linkbase is a database.

387. The method of claim 383, wherein said linkbase operates as a process.

388. The method of claim 383, wherein the interactive television state information corresponding to a particular device includes an indication of channel.

389. The method of claim 383, wherein the interactive television state information corresponding to a particular device includes an indication of broadcast time.

390. The method of claim 383, wherein the correlated state information includes indications of channels.

391. The method of claim 383, wherein the correlated state information includes indications of broadcast times.

392. The method of claim 383, wherein said request is dispatched via a computer network.

393. The method of claim 392, wherein said network is the internet.

394. A method of operating a media player, comprising:

having a computer create a state record relating to a session at said computer; and

establishing a session at the media player, the media player session based on said state record.

395. The method of claim 394, wherein said media player and said computer do not share operational logic.

396. The method of claim 394, wherein the media player session presents the ending resource for a link activated at the computer.

397. The method of claim 394, wherein the media player is a digital video recorder.

398. The method of claim 394, wherein the media player is a television set.

399. The method of claim 394, wherein said computer is a personal computer.

400. The method of claim 394, wherein said computer is a personal data assistant.

401. A method of operating a computer, comprising:

creating a state record relating to a session at said computer; and

having a media player establish a session based on said state record.

402. The method of claim 401, wherein said media player and said computer do not share operational logic.

403. The method of claim 401, wherein the media player session presents the ending resource for a link activated at the computer.

404. A method of operating a media player, comprising:

having a media player create a state record containing state information relating to the media player; and

establishing a session at a computer, said session based on said state information.

405. The method of claim 404, wherein said media player and said computer do not share operational logic.

406. The method of claim 404, wherein the media player is a digital video recorder.

407. The method of claim 404, wherein said media player is a television.

408. The method of claim 404, wherein said computer is a personal computer.

409. The method of claim 404, wherein said computer is a personal data assistant.

410. The method of claim 404, wherein the state information is used at the computer system to identify a hyperlink ending resource for said session.

411. A method of conducting a browsing session, comprising:

obtaining, at an independent system collocated with a media player, information relating to a program currently being viewed at said media player, wherein the information is obtained via a wide area network;

using the information to determine a source for enhancement resources related to that program; and

presenting the enhancement resources on the independent system.

412. A method of recording events at a device, comprising:

recording in a log, at a desired level of granularity, a device's events;

identifying recorded events that have been made superfluous by changes in the device's state; and

deleting the identified events from the log.

413. The method of claim 412, wherein the device uses a model-view-controller architecture.

414. The method of claim 412, wherein the recording of events is used for a one-time process of replicating state.

415. The method of claim 414, wherein the one-time process is the initiation of a synchronization process which then continues.

416. The method of claim 415, wherein the synchronization is in support of collaboration by multiple users.

417. The method of claim 414, wherein the one-time process is the transfer of a multi-machine user interface session.

418. The method of claim 412, wherein the events are used to synchronize activity among the active members of a group of systems whose membership may vary.

419. The method of claim 418, wherein said deleting is limited to related changes that occur within synchronization intervals bounded by points in time when membership in the active group changes.

420. A method of providing interactive television services, comprising:

enabling a user at a television device to conduct a user session, said user session requiring services resident on an independent user computer; and

providing said device access to said services.

421. The method of claim 420, wherein said services are provided as web services using simple object access protocol.

422. The method of claim 420, wherein the computer is one of a personal computer, a personal data assistant, and a tablet

423. The method of claim 421, wherein the web services relate to one of personal productivity information and financial information.

424. The method of claim 423, wherein said personal productivity information comprises at least one of contact lists, calendars, and mail logs.

425. The method of claim 423, wherein said financial information comprises at least one of payables, receivables, stock transactions, stock portfolios, banking transactions, banking holdings, and shopping.

426. The method of claim 420, wherein the method is conducted at one of the location of the television and a remote location.

427. The method of claim 426, wherein said remote location is a head-end.

428. A method of providing venue-based computer-related services comprising:

employing a venue-based device set to offer a programmatic service to a user-supplied computer collocated with said device set;

having the device set provide said service to the computer, wherein the computer's user interacts with the venue-based device set while the device set draws on the resources of the computer.

429. A method for providing session portability, comprising:

maintaining state information corresponding to a session being carried out on a first device;

transferring elements of said state information to a second device; and

using said elements to reestablish said session at said second device.

430. The method of claim 429, wherein said session is a browser session.

431. The method of claim 429, wherein the state information is provided, at the time of a request for said transfer, by an application supporting the session.

432. The method of claim 429, wherein said first device is a media player and said second device is a computer-type device.

433. The method of claim 429, wherein said first device is a computer-type device and said second device is a media player.

434. The method of claim 429, wherein said first device is a venue-based device and said second device is a user-supplied computer.

435. The method of claim 429, wherein said first device is a user-supplied computer and said second device is a venue-based device.

436. A method for establishing coupled sessions, comprising:

assembling a first state record corresponding to a first session being carried out on a first device;

transferring elements of said state record to a second device;

using said elements to establish a second session at said second device;

assembling second state record corresponding to said second session; and

utilizing at least portions of said state records in the maintenance of synchrony between said sessions.

437. The method of claim 436, wherein the assembling of said first state record is initiated in response to a request for said transfer.

438. The method of claim 437, wherein said request is submitted after a series of user interactions.

439. The method of claim 436, wherein first and second devices are used in combination by a user to function as a multi-machine user interface.

440. The method of claim 436, wherein said first device is a media player and said second device is a computer.

441. The method of claim 436, wherein said first device is a computer and said second device is a media player.

442. The method of claim 436, wherein said first device is a venue-based device and said second device is a user-supplied computer.

443. The method of claim 436, wherein said first device is a user-supplied computer and said second device is a venue-based device.

444. A method for hyperlink traversal, comprising:

maintaining state information corresponding to a session being carried out on a first device;

transferring elements of said state information to a second device, said portion including information relating to a hyperlink; and

using said elements to establish at said second device a modified version of said session, the modification being in accordance with said hyperlink.

445. The method of claim 444, wherein said first device is a media player and said second device is a computer-type device.

446. The method of claim 444, wherein said first device is a computer-type device and said second device is a media player.

447. The method of claim 444, wherein said first device is a venue-based device and said second device is a user-supplied computer.

448. The method of claim 444, wherein said first device is a user-supplied computer and said second device is a venue-based device.

449. A method for operating a modular computer, comprising:

presenting resources while the core module of said computer is connected to a first device set;

disconnecting said core module from said first device set;

connecting said core module to a second device set; and

while said core module is connected to said second device set, automatically presenting said resources in a manner that adapts said resources to the characteristics of said second device set.

450. A system for providing a session, comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

receiving a state record containing state information relating to a media player; and

establishing a session at said system, said session based on said state information.

451. The system of claim 450, wherein said media player and said system do not share operational logic.

452. The system of claim 450, wherein the media player is a digital video recorder.

453. The system of claim 450, wherein said media player is a television.

454. The system of claim 450, wherein said system is a personal computer.

455. The system of claim 450, wherein said system is a personal data assistant.

456. The system of claim 450, wherein the state record is used at the system to identify a hyperlink starting resource for said session.

457. The system of claim 450, wherein said media player and said system use at least one of separate hardware logic instances and separate software process instances.

458. The system of claim 450, wherein a user may, while utilizing a session at said media player, employ a navigation control to select said system.

459. The system of claim 458, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be the ending resource for a link activated at the first device set.

460. The system of claim 450, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be a resource currently presented at the first device set.

461. The system of claim 450, wherein said state record is accessible to said system.

462. The system of claim 450, wherein said processor further performs the step of converting a resource corresponding to a session at said media player into a format suitable for presentation at said system.

463. The system of claim 450, wherein said media player is one of a browser and an interactive television viewing system.

464. The system of claim 463, wherein the browser is a web browser.

465. The system of claim 450, wherein navigation actions at said system set can target said media player.

466. The system of claim 450, wherein said media player comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, and a personal computer acting as a media gateway/server.

467. The system of claim 450, wherein said system comprises one of a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

468. The system of claim 450, wherein a resource presented at said system is the ending resource for a link activated at said media player.

469. The system of claim 468, wherein the media player is presenting a television-type program, and the ending resource comprises enhancement content associated with the television-type program.

470. The system of claim 469, wherein a user may choose to target some enhancement content resources to the media player, and other enhancement content resources to the system.

471. The system of claim 468, wherein the ending resource comprises one of enhancement content associated with the television-type program, and an alternative television program resource and time position.

472. The system of claim 450, wherein an interactive television session is associated with said media player.

473. The system of claim 450, wherein a session relating to simple television viewing is presented at said media player, and the session associated with said system relates to enhancement content associated with television viewing.

474. The system of claim 450, wherein the system set is activated just prior to said establishing, without need for said media player to be specifically pre-configured to interact with said system.

475. The system of claim 474, wherein one of a device set resource name and a device set network address is resolved dynamically.

476. The system of claim 450, wherein the system is activated after said establishing without need for said media player to be specifically pre-configured to interact with said system.

477. The system of claim 450, wherein one of a device set resource name and a device set network address is resolved dynamically.

478. The system of claim 450, wherein said state information comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

479. The system of claim 478, wherein said state information comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across said media player and said system.

480. The system of claim 478, wherein a cookie is sent to a remote server from each of the media player and the system in order to maintain a remote session with said server.

481. The system of claim 480, wherein the remote session employs one of HTTP and HTTPS.

482. The system of claim 480, wherein, through the use of a proxy server, said remote server views said media player and said system as having identical network identities.

483. The system of claim 482, wherein the actual network identity corresponding to said media player is not identical to the actual network identity corresponding to said system.

484. The system of claim 482, wherein said network identities comprise network addresses.

485. The system of claim 482, wherein said proxy server is collocated with said media player.

486. The system of claim 450, wherein said state record comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

487. The system of claim 486, wherein said state record comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across said media player and said system.

488. The system of claim 486, wherein a cookie is sent to a remote server from each of the media player and the system in order to maintain a remote session with said server.

489. The system of claim 488, wherein remote session employs one of HTTP and HTTPS.

490. The system of claim 488, wherein, through the use of a proxy server, said remote server views said media player and said system as having identical network identities.

491. The system of claim 490, wherein the actual network identity corresponding to said media player is not identical to the actual network identity corresponding to said system.

492. The system of claim 490, wherein said network identities comprise network addresses.

493. The system of claim 490, wherein said proxy server is collocated with said media player.

494. The system of claim 450, wherein at least one of said system and said media player has a form factor suited to across-the-room viewing and readily viewed by multiple users, and at least one of said system and said media player has a form factor suited to close work by a single user.

495. The system of claim 450, wherein said processor further performs the step of:

additionally establishing a session at a device set, said session based on said state information,

wherein said system and said device set are allocated for use by different users.

496. The system of claim 495, wherein said system and said device set are used independently.

497. The system of claim 495, wherein said different users perform navigation actions that target the media player.

498. . The system of claim 450, wherein rules limit the ability of a user to target a multi-user device set.

499. The system of claim 450, wherein multiuser collaboration features are supported.

500. The system of claim 450, wherein said media player and said system are coordinated by a central controller.

501. The system of claim 450, wherein said media player and said system sets are coordinated on a peer-to-peer basis without a central controller.

502. The system of claim 450, wherein a remote session state, associated with a remote system in communication with the media player, is further associated with said system.

503. The system of claim 502, wherein said media player and said system have different network addresses.

504. The system of claim 503, wherein said network addresses comprise IP addresses.

505. The system of claim 450, wherein said establishing occurs during an intermediate step in a remote session comprising the processing of a transaction with a remote server, wherein the continuity of said remote session is maintained.

506. The system of claim 450, wherein a user may set preferences that specify a default targeting of links to device sets.

507. The system of claim 506, wherein the user preferred default target is selectively specified in terms of at least one of: types of starting resources, types of ending resources, metadata attributes of starting resources, metadata attributes of ending resources, attributes of links, application categories, and defined alternative navigation modes.

508. The system of claim 450, wherein a resource presented at said media player may contain recommended targeting behavior attributes that specify a default preferred targeting of links to classes of device sets.

509. The system of claim 506, wherein said default targeting is specified in terms of at least one of defined alternative navigation modes, and the active states of the device sets.

510. The system of claim 450, wherein a link to an ending resource is specified, external to a starting resource presented to said media player, as a third party arc.

511. The system of claim 450, wherein a starting resource presented to said media player is one of 1) a resource containing links as outbound arcs, 2) a resource associated with external third-party arcs, 3) a resource containing coding other than links that specifies associations to other resources that give the effect of links, and 3) a resource associated with external coding that specifies associations to other resources that give the effect of links.

512. The system of claim 511, wherein said starting resource is distributed with associated third party arcs.

513. The system of claim 511, wherein an independent association is made with third-party arcs that are sourced separately from the starting resource.

514. The system of claim 511, wherein said third party arcs are associated with an entire linear resource.

515. The system of claim 512, wherein said third party arcs are associated with a time span for a continuous media resource.

516. The system of claim 512, wherein said third party arcs are associated with a region of an image resource.

517. The system of claim 511, wherein said coding other than links includes at least one of a menu structure having a defined graph structure, a transaction request form having a defined address for transaction submission, and a selectable content element having a defined address for related action.

518. The system of claim 450, wherein said establishing comprises targeting a defined display subunit of said system, said display subunit comprising one of a screen area, a pane, a frame, a window, a picture-in-picture region, a physical subunit, a virtual subunit, and a relevant presentation context.

519. The system of claim 518, wherein said establishing may further comprise specification of attributes to select among full-screen, a specified intermediate size and position, a previous intermediate size and position, and a minimized display on said system.

520. The system of claim 450, wherein a resource presented is obtained from at least one of a persistent storage device and a communications channel.

521. The system of claim 520, wherein said persistent storage device and said communications channel are connected via one of a) one of said media player and said system, and b) each of said media player and said system.

522. The system of claim 450, wherein coordination between said media player and said system is effected using communications paths that link one of a) said media player and said system directly to one another, b) said media player and said system via an intermediate gateway system serving both, c) one of said media player and said system via an intermediate gateway system serving the one of said media player and said system, d) a local control system serving one of said media player and said system , e) a local control system serving both said media player and said system, f) a remote control system serving one of said media player and said system, and g) a remote control system serving both of said media player and said system.

523. The system of claim 450, wherein said media player and said system share network access facilities.

524. The system of claim 450, wherein said media player and said system are independently provisioned.

525. The system of claim 450, wherein at least one of internet protocol, digital television, Internet, cable, satellite, broadcast, mobile, and digital versatile disk is employed.

526. The system of claim 450, wherein resources and links associated with said session are defined by at least one of hypertext markup language, XML, extensible markup language linking language, motion picture experts group specifications, advanced television enhancement forum specifications, and synchronized multimedia integration language.

527. The system of claim 450, wherein said system is selected based upon on one of a) attributes ordinarily used to control the targeting of navigation using a single display device, and b) attributes specific to presentation using multiple device sets.

528. The system of claim 450, wherein said second device set is selected based upon attributes comprising one of a) a new presentation context, b) a replacement presentation context that loads the presentation of an ending resource in the same context in which the presentation of a starting resource was loaded, c) an embedded presentation context that loads the presentation of an ending resource in place of a starting resource.

529. The system of claim 450, wherein said establishing comprises employing as a presentation context one or more of a screen area, a pane, a frame, a window, and a picture-in-picture region.

530. The system of claim 527, wherein said attributes ordinarily used to control the targeting of navigation using a single display device are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

531. The system of claim 527, wherein said attributes specific to presentation using multiple device sets are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

532. The system of claim 530, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

533. The system of claim 531, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

534. The system of claim 530, wherein said hypertext markup language attributes are hypertext markup language target attributes.

535. The system of claim 531, wherein said hypertext markup language attributes are hypertext markup language target attributes.

536. The system of claim 530, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

537. The system of claim 531, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

538. The system of claim 530, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

539. The system of claim 531, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

540. The system of claim 450, wherein said session is presented in accord with one of extensible markup language linking language actuate attributes hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes, and extensible hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes.

541. The system of claim 450, wherein said system is selected such that there is no change in selection preference settings that affect subsequent link navigation steps unless a user performing the selection indicates that there be a change in selection preference settings that affect subsequent link navigation steps.

542. The system of claim 450, wherein a user preferred default selection of said system is pre-specified by a third party, including a system component provider.

543. The system of claim 450, wherein said establishing is triggered by a user.

544. The system of claim 450, wherein said establishing is triggered at any of multiple stages in a session associated with said media player.

545. The system of claim 450, wherein said system is selected from computers of varying form factors, wherein resources associated with a session associated with said media player are altered only with regard to presentation adaptations needed to adapt said resources to the form factor of said system.

546. The system of claim 545, wherein said adaptations relate to at least one of display parameters and input device parameters.

547. The system of claim 546, wherein said display parameters comprise at least one of size, resolution, and color.

548. The system of claim 546, wherein said input device parameters comprise at least one of keyboard, stylus, touch screen, mouse, and buttons.

549. The system of claim 450, wherein said establishing maintains user state at said media player.

550. The system of claim 450, wherein said establishing maintains server state for a server that was allied, at the time of said establishing, with a session associated with said media player. at the time of continuing.

551. The system of claim 450, wherein state continuation after said establishing is supported at a granularity corresponding to completed user interaction steps and user-visible results.

552. The system of claim 551, wherein said state continuation does not maintain details of internal program execution state of software processes that respond to a user's interaction but are not visible to the user.

553. The system of claim 450, wherein state continuation after said establishing is supported at a granularity corresponding to the identity, time-position and presentation state of a browsed resource.

554. The system of claim 553, wherein said state continuation need not maintain details of user interaction steps regarding data entry inputs to be sent to a resource.

555. The system of claim 450, wherein state continuation after said establishing is supported at a granularity corresponding to the identity of a browsed resource.

556. The system of claim 555, wherein said state continuation need not maintain details of time position and user interaction steps regarding a resource.

557. The system of claim 450, wherein an interactive television session is presented at said media player, and said system is at one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

558. The system of claim 450, where said media player is a lean-back configuration, and said system is a lean-forward configuration.

559. The system of claim 450, wherein said establishing is triggered at a loading of a content resource.

560. The system of claim 450, wherein said establishing is triggered at a traversal of a link.

561. The system of claim 450, wherein said establishing is triggered at an intermediate state in the play of a continuous media content resource.

562. The system of claim 450, wherein said establishing is triggered when a user interaction with the presentation of a content resource is complete.

563. The system of claim 562, wherein said content resource is a form.

564. The system of claim 563, wherein said user interaction is the filling out of said form.

565. The system of claim 450, wherein resources accessed in said session remain independent of said establishing.

566. The system of claim 545, wherein said adaptations are accommodated by adaptations performed at said system after said system accesses said resources.

567. The system of claim 545, wherein said adaptations are accommodated by adaptations performed, at a remote server, after an access to said resources executed by said system.

568. The system of claim 545, wherein no intermediary server is used to adapt said resources.

569. The system of claim 566, wherein said accommodation is guided by coding, associated with said resources, that specifies recommended accommodations corresponding to varying form factors.

570. The system of claim 450, wherein said processor further performs the step of moving said session from said system back to said media player.

571. The system of claim 450, wherein said establishing is independent of further actions at said media player.

572. The system of claim 571, wherein a session relating to said media player is forked into a first session corresponding to said media player and a second session corresponding to said system, the two sessions utilizing two separate browser instances.

573. The system of claim 572, wherein the separate browser instance corresponding to one of said media player and said system is not affected by further browsing actions at the other of said media player and said system unless requested by a user.

574. The system of claim 450, wherein said session is additional to a session already established at said system.

575. The system of claim 450, wherein said session replaces a specified session at said system.

576. The system of claim 571, wherein a targeting of one of said media player and said system from the other of said media player and said system is designated to trigger a further establishment.

577. The system of claim 576, wherein the further establishment is designated to do one of 1) creating a new session at the targeted one of said media player and said system that is additional to an existing session there, and 2) replacing the current state of a specified session at the targeted one of said media player and said system.

578. The system of claim 450, wherein said establishing involves one of 1) moving all browser instances associated with a session allied with said media player, and 2) moving only the browser instance from which said establishing is initiated.

579. The system of claim 450, wherein said establishing is triggered by activating said system.

580. The system of claim 450, wherein said establishing is triggered by deactivating said media player.

581. The system of claim 450, Wherein said state information is communicated from a browser instance at media player set to a browser instance at said system set at the time of establishing.

582. The system of claim 581, wherein state information changing events prior to the establishment are not individually logged.

583. The system of claim 581, wherein, and said state record is single state record that defines the current state information relating to said media player at said time.

584. The system of claim 450, wherein a simple event history, in the form of a history list that records the identity and sequence of resources navigated and a current list of the recent

sequence of that resource navigation, is provided in said state record as a current state record element.

585. The system of claim 450, wherein the system is employable by browser processes differing by one of different software code and different hardware architectures.

586. The system of claim 585, wherein said browser processes are capable of exporting and importing said state record using a defined common format.

587. The system of claim 586, wherein the system and the defined common format is made available for use by any provider of browsers under reasonable and non-discriminatory terms.

588. The system of claim 450, wherein said establishing is triggered by a user transfer request signal.

589. The system of claim 450, wherein said establishing is triggered by one of navigation to a triggering resource and navigation of a triggering link.

590. The system of claim 589, wherein the triggering resource is a resource belonging to a category pre-defined to cause triggering.

591. The system of claim 589, wherein the triggering resource is a resource containing a specific attribute specifying a trigger.

592. The system of claim 589, wherein the category pre-defined to cause triggering is one of a navigation level, a content type, and an activity type.

593. The system of claim 582, wherein maintenance of information needed to transfer cumulative state is routinely performed.

594. The system of claim 593, wherein said maintenance is routinely performed once support for said media player and said system is first enabled at one of system setup time and device set setup time.

595. The system of claim 593, wherein said maintenance does not depend on a user activating a special state tracking activity for said session.

596. The system of claim 450, wherein said processor further performs the steps of:

said media player providing user identification information corresponding to a user; and

said system providing user identification information corresponding to said user.

597. The system of claim 596, wherein user authentication information is provided by both said media player and said system.

598. The system of claim 450, wherein said state information is used at said system to identify a resource for browsing.

599. The system of claim 598, wherein the resource is used as a hyperlink starting resource with one of a designated set of third-party arcs to determine a hyperlink ending resource for traversal at said system.

600. The system of claim 598, wherein said state record comprises a current time position and a current time.

601. The system of claim 598, wherein said establishing is triggered by a user signal.

602. The system of claim 598, wherein establishing is performed when a program change event occurs.

603. The system of claim 599, wherein the hyperlink ending resource is presented to a user for activation.

604. The system of claim 450, wherein said processor further performs the step of:

transmitting said state record from said media player to said system, wherein said media player set initiates the transmission.

605. The system of claim 450, wherein said processor further performs the step of:

said system requesting said state record.

606. The method of claim 450, wherein said processor further performs the steps of:

acquiring information of the identity of a resource associated with a session presented at said media player; and

employing said information to determine a source for enhancement resources related to said session presented at said media player.

607. The system of claim 606, wherein the information is obtained via a wide area network.

608. The system of claim 607, wherein the wide area network is the Internet.

609. The system of claim 450, wherein the state record includes a navigation history, wherein said navigation history can be retraced as if said continuing had not occurred.

610. A system for providing a browsing session comprising:

a first memory having program code stored therein; and

a first processor operatively connected to said first memory for carrying out instructions in accordance with the stored program code of said first processor;

wherein the program code of said first processor, when executed by said first processor, causes said first processor to perform the step of:

beginning a session comprising browsing interactions at a first device set; and

a second memory having program code stored therein; and

a second processor operatively connected to said second memory for carrying out instructions in accordance with the stored program code of said second processor;

wherein the program code of said second processor, when executed by said second processor, causes said second processor to perform the step of:

continuing said session at a second device set.

611. The system of claim 610, wherein a resource presented at the second device set is the ending resource for a link activated at the first device set.

612. The system of claim 610, wherein said first device set and said second device set use different browser instances.

613. The system of claim 610, wherein said first device set and said second device set use at least one of separate processors and separate software process instances.

614. The system of claim 610, wherein a user may, while utilizing said session at said first device set, employ a navigation control to select said second device set.

615. The system of claim 614, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be the ending resource for a link activated at the first device set.

616. The system of claim 614, wherein a user may employ said navigation control to indicate that a resource presented at the second device set should be a resource currently presented at the first device set.

617. The system of claim 610, wherein state data corresponding to said session at said first device set is provided to said second device set.

618. The system of claim 610, wherein state data corresponding to said session at said first device set is accessible to said second device set.

619. The system of claim 610, wherein said processor further performs the step of converting a resource corresponding to said session at said first device into a format suitable for presentation at said second device set.

620. The system of claim 610, wherein said second device is one of a browser and an interactive television viewing system.

621. The system of claim 620, wherein the browser is a web browser.

622. The system of claim 610, wherein navigation actions at the second device set can target the first device set.

623. The system of claim 610, wherein said first device set comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

624. The system of claim 610, wherein said second device set comprises one of a television, a television plus set-top box, a television plus media gateway/server, a digital video recorder, a personal computer acting as a television, a personal computer acting as a media gateway/server, a personal computer, a remote control relying on intelligence in another device, an independent remote control, a personal data assistant, a tablet, a touchpad, an Internet appliance, and a wireless phone.

625. The system of claim 610, wherein said first device set and said second device set are of dissimilar types.

626. The system of claim 611, wherein the first device set is a television-type device set presenting a television-type program, the second device set is a computer-type device set, and the ending resource comprises enhancement content associated with the television-type program.

627. The system of claim 626, wherein a user may choose to target some enhancement content resources to the television-type device, and other enhancement content resources to the personal computer-type device.

628. The system of claim 611, wherein the second device set is a television-type device set presenting at least one of a television-type program and enhancements, the first device set is a computer-type device set, and the ending resource comprises one of enhancement content associated with the television-type program, and an alternative television program resource and time position.

629. The system of claim 610, wherein said session comprises an interactive television session, the first device set is a television, and the second device set is one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

630. The system of claim 610, wherein the first device set is a lean-back configuration, and the second device set is a lean-forward configuration.

631. The system of claim 610, wherein said session as presented at said first device set relates to simple television viewing, and said session as presented at said second device set relates to enhancement content associated with television viewing.

632. The system of claim 610, wherein the second device set is activated just prior to said continuing, without need for said first device set to be specifically pre-configured to interact with said second device set.

633. The system of claim 632, wherein one of a device set resource name and a device set network address is resolved dynamically.

634. The system of claim 610, wherein the second device set is activated after said continuing without need for said first device set to be specifically pre-configured to interact with said second device set.

635. The system of claim 634, wherein one of a device set resource name and a device set network address is resolved dynamically.

636. The system of claim 617, wherein said state data comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

637. The system of claim 636, wherein said state data comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across the two device sets.

638. The system of claim 636, wherein a cookie is sent to a remote server from each of the first and second device sets in order to maintain a remote session with said server.

639. The system of claim 638, wherein the remote session employs one of HTTP and HTTPS.

640. The system of claim 638, wherein, through the use of a proxy server, said remote server views said first device set and said second device set as having identical network identities.

641. The system of claim 640, wherein the actual network identity corresponding to said first device set is not identical to the actual network identity corresponding to said second device set.

642. The system of claim 640, wherein said network identities comprise network addresses.

643. The system of claim 640, wherein said proxy server is collocated with said first device set.

644. The system of claim 618, wherein said state data comprises at least one of a current path, a history list, a cookie, a set of forms contents, a defined remote session state record, and identification and authentication information.

645. The system of claim 644, wherein said state data comprises at least identification and authentication information, and wherein said identification and authentication information establishes a continuing identity across the two device sets.

646. The system of claim 644, wherein a cookie is sent to a remote server from each of the first and second device sets in order to maintain a remote session with said server.

647. The system of claim 646, wherein remote session employs one of HTTP and HTTPS.

648. The system of claim 646, wherein, through the use of a proxy server, said remote server views said first device set and said second device set as having identical network identities.

649. The system of claim 648, wherein the actual network identity corresponding to said first device set is not identical to the actual network identity corresponding to said second device set.

650. The system of claim 648, wherein said network identities comprise network addresses.

651. The system of claim 648, wherein said proxy server is collocated with said first device set.

652. The system of claim 610, wherein at least one of said device sets has a form factor suited to across-the-room viewing and readily viewed by multiple users, and at least one of said device sets has a form factor suited to close work by a single user.

653. The system of claim 610, wherein said processor further performs the step of:

additionally continuing said session at a third device set,

wherein said second device set and said third device set are allocated for use by different users.

654. The system of claim 653, wherein said second device set and said third device set are used independently.

655. The system of claim 653, wherein said different users perform navigation actions that target the first device set.

656. The system of claim 610, wherein rules limit the ability of a user to target a multi-user device set.

657. The system of claim 610, wherein multiuser collaboration features are supported.

658. The system of claim 610, wherein said device sets are coordinated by a central controller.

659. The system of claim 610 wherein said device sets are coordinated on a peer-to-peer basis without a central controller.

660. The system of claim 610, wherein a remote session state, associated with a remote system in communication with the first device set, is further associated with said second device set.

661. The system of claim 660, wherein the first and second device sets have different network addresses.

662. The system of claim 661, wherein said network addresses comprise IP addresses.

663. The system of claim 610, wherein said continuing occurs during an intermediate step in a remote session comprising the processing of a transaction with a remote server, wherein the continuity of said remote session is maintained.

664. The system of claim 610, wherein a user may set preferences that specify a default targeting of links to device sets.

665. The system of claim 664, wherein the user preferred default target is selectively specified in terms of at least one of: types of starting resources, types of ending resources, metadata attributes of starting resources, metadata attributes of ending resources, attributes of links, application categories, and defined alternative navigation modes.

666. The system of claim 610, wherein a resource presented at said first device set may contain recommended targeting behavior attributes that specify a default preferred targeting of links to classes of device sets.

667. The system of claim 664, wherein said default targeting is specified in terms of at least one of defined alternative navigation modes, and the active states of the device sets.

668. The system of claim 610, wherein a link to an ending resource is specified, external to a starting resource presented to said first device set, as a third party arc.

669. The system of claim 610, wherein a starting resource presented to said first device set is one of 1) a resource containing links as outbound arcs, 2) a resource associated with external third-party arcs, 3) a resource containing coding other than links that specifies associations to other resources that give the effect of links, and 3) a resource associated with external coding that specifies associations to other resources that give the effect of links.

670. The system of claim 669, wherein said starting resource is distributed with associated third party arcs.

671. The system of claim 669, wherein an independent association is made with third-party arcs that are sourced separately from the starting resource.

672. The system of claim 669, wherein said third party arcs are associated with an entire linear resource.

673. The system of claim 670, wherein said third party arcs are associated with a time span for a continuous media resource.

674. The system of claim 670, wherein said third party arcs are associated with a region of an image resource.

675. The system of claim 669, wherein said coding other than links includes at least one of a menu structure having a defined graph structure, a transaction request form having a defined address for transaction submission, and a selectable content element having a defined address for related action.

676. The system of claim 610, wherein said continuing comprises targeting a defined display subunit of said second device set, said display subunit comprising one of a screen area, a pane, a frame, a window, a picture-in-picture region, a physical subunit, a virtual subunit, and a relevant presentation context.

677. The system of claim 676, wherein said continuing may further comprise specification of attributes to select among full-screen, a specified intermediate size and position, a previous intermediate size and position, and a minimized display on said second device set.

678. The system of claim 610, wherein a resource presented in conjunction with said session is obtained from at least one of a persistent storage device and a communications channel.

679. The system of claim 678, wherein said persistent storage device and said communications channel are connected via one of a) one of the device sets, and b) each of the devices sets.

680. The system of claim 610, wherein coordination between the device sets is effected using communications paths that link one of a) the device sets directly to one another, b) the device sets via an intermediate gateway system serving both, c) one of the device sets via an intermediate gateway system serving that device set, d) a local control system serving one of said device sets, e) a local control system serving both of said device sets, f) a remote control system serving one of said device sets, and g) a remote control system serving both of said device sets.

681. The system of claim 610, wherein each of said device sets has the capability of operating as a browser entirely independently of the other of said device sets.

682. The system of claim 681, wherein said device sets share network access facilities.

683. The system of claim 681, wherein the device sets are independently provisioned.

684. The system of claim 610, wherein at least one of internet protocol, digital television, Internet, cable, satellite, broadcast, mobile, and digital versatile disk is employed.

685. The system of claim 610, wherein resources and links associated with said session are defined by at least one of hypertext markup language, extensible markup language, extensible markup language linking language, Motion Picture Experts Group specifications, advanced television enhancement forum specifications, and synchronized multimedia integration language.

686. The system of claim 610, wherein said second device set is selected based upon on one of a) attributes ordinarily used to control the targeting of navigation using a single display device, and b) attributes specific to presentation using multiple device sets.

687. The system of claim 610, wherein said second device set is selected based upon attributes comprising one of a) a new presentation context, b) a replacement presentation context that loads the presentation of an ending resource in the same context in which the presentation of a starting resource was loaded, c) an embedded presentation context that loads the presentation of an ending resource in place of a starting resource.

688. The system of claim 610, wherein said continuing comprises employing as a presentation context one or more of a screen area, a pane, a frame, a window, and a picture-in-picture region.

689. The system of claim 686, wherein said attributes ordinarily used to control the targeting of navigation using a single display device are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

690. The system of claim 686, wherein said attributes specific to presentation using multiple device sets are one of extensible markup language linking language attributes, hypertext markup language attributes, extensible hypertext markup language attributes, and synchronized multimedia integration language attributes.

691. The system of claim 689, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

692. The system of claim 690, wherein said extensible markup language linking language attributes are extensible markup language linking language show attributes.

693. The system of claim 689, wherein said hypertext markup language attributes are hypertext markup language target attributes.

694. The system of claim 690, wherein said hypertext markup language attributes are hypertext markup language target attributes.

695. The system of claim 689, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

696. The system of claim 690, wherein said extensible hypertext markup language attributes are extensible hypertext markup language target attributes.

697. The system of claim 689, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

698. The system of claim 690, wherein said synchronized multimedia integration language attributes are synchronized multimedia integration language target attributes.

699. The system of claim 610, wherein said session is presented in accord with one of extensible markup language linking language actuate attributes hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes, and extensible hypertext markup language attributes equivalent to extensible markup language linking language actuate attributes.

700. The system of claim 610, wherein said second device set is selected such that there is no change in selection preference settings that affect subsequent link navigation steps unless a

user performing the selection indicates that there be a change in selection preference settings that affect subsequent link navigation steps.

701. The system of claim 610, wherein said second device set is selected such that said second device set is assumed to be the first device set unless indicated otherwise by one of user action at the time of selection, user preferences established prior to the time of selection, and content author recommendations.

702. The system of claim 610, wherein a user preferred default selection of said second device set is pre-specified by a third party, including a system component provider.

703. The system of claim 610, wherein said continuing is triggered by a user.

704. The system of claim 610, wherein said continuing is triggered at any of multiple stages in said session.

705. The system of claim 610, wherein said second device set is selected from device sets of varying form factors, wherein resources associated with said session are altered only with regard to presentation adaptations needed to adapt said resources to the form factor of said second device set.

706. The system of claim 705, wherein said adaptations relate to at least one of display parameters and input device parameters.

707. The system of claim 706, wherein said display parameters comprise at least one of size, resolution, and color.

708. The system of claim 706, wherein said input device parameters comprise at least one of keyboard, stylus, touch screen, mouse, and buttons.

709. The system of claim 610, wherein said continuing maintains user state at said first device set.

710. The system of claim 610, wherein said continuing maintains server state for a server that was associated with said session at the time of continuing.

711. The system of claim 610, wherein state continuation after said continuing is supported at a granularity corresponding to completed user interaction steps and user-visible results.

712. The system of claim 711, wherein said state continuation does not maintain details of internal program execution state of software processes that respond to a user's interaction but are not visible to the user.

713. The system of claim 610, wherein state continuation after said continuing is supported at a granularity corresponding to the identity, time-position and presentation state of a browsed resource.

714. The system of claim 713, wherein said state continuation need not maintain details of user interaction steps regarding data entry inputs to be sent to a resource.

715. The system of claim 610, wherein state continuation after said continuing is supported at a granularity corresponding to the identity of a browsed resource.

716. The system of claim 715, wherein said state continuation need not maintain details of time position and user interaction steps regarding a resource.

717. The system of claim 610, wherein the session comprises an interactive television session, and said first device is a television and said second device is at one of a personal computer, a personal data assistant, a tablet, a remote control, and a mobile phone.

718. The system of claim 610, wherein said continuing is triggered at a loading of a content resource.

719. The system of claim 610, wherein said continuing is triggered at a traversal of a link.

720. The system of claim 610, wherein said continuing is triggered at an intermediate state in the play of a continuous media content resource.

721. The system of claim 610, wherein said continuing is triggered when a user interaction with the presentation of a content resource is complete.

722. The system of claim 721, wherein, said content resource is a form.

723. The system of claim 722, wherein said user interaction is the filling out of said form.

724. The system of claim 610, wherein resources accessed in said session remain independent of said continuing.

725. The system of claim 705, wherein said adaptations are accommodated by adaptations performed at said second device set after said second device set accesses said resources.

726. The system of claim 705, wherein said adaptations are accommodated by adaptations performed, at a remote server, after an access to said resources executed by said second device set.

727. The system of claim 705, wherein no intermediary server is used to adapt said resources.

728. The system of claim 725, wherein said accommodation is guided by coding, associated with said resources, that specifies recommended accommodations corresponding to varying form factors.

729. The system of claim 610, wherein said processor further performs the step of moving said session from said second device set back to said first device set.

730. The system of claim 610, wherein said continuing is independent of further actions at said first device set.

731. The system of claim 729, wherein said session is forked into a first session corresponding to said first device set and a second session corresponding to said second device set, the two sessions utilizing two separate browser instances.

732. The system of claim 731 wherein the separate browser instance corresponding to one of said device sets is not affected by further browsing actions at the other of said device sets unless requested by a user.

733. The system of claim 730, wherein said continuing is designated to do one of 1) creating a new session at the second device set that is additional to an existing session there, and 2) replacing the current state of a specified session at the second device set.

734. The system of claim 730, wherein a targeting of one of said device sets from the other of said device sets is designated to trigger a further continuance.

735. The system of claim 734, wherein the further continuance is designated to do one of 1) creating a new session at the targeted device set that is additional to an existing session there, and 2) replacing the current state of a specified session at said targeted device set.

736. The system of claim 610, wherein said continuing involves one of 1) moving all browser instances associated said session as presented at said first device set, and 2) moving only the browser instance from which said continuing is initiated.

737. The system of claim 610, wherein said continuing is triggered by activating the second device set.

738. The system of claim 610, wherein said continuing is triggered by deactivating the first device set.

739. The system of claim 610, Wherein the state of the session is communicated from a browser instance at the first device set to a browser instance at the second device set at the time of continuing.

740. The system of claim 739, wherein session state change events prior to the continuation are not individually logged.

741. The system of claim 739, wherein the state of the session is transferred as a single state record that defines the current cumulative state of the session at said time.

742. The system of claim 740, wherein a simple event history, in the form of a history list that records the identity and sequence of resources navigated and a current list of the recent sequence of that resource navigation, is provided in the state record as a current state record element.

743. The system of claim 610, wherein the system is employable by browser processes differing by one of different software code and different hardware architectures.

744. The system of claim 743, wherein said browser processes are capable of exporting and importing a state record corresponding to said session using a defined common format.

745. The system of claim 744, wherein the system and the defined common format is made available for use by any provider of browsers under reasonable and non-discriminatory terms.

746. The system of claim 610, wherein said continuing is triggered by a user transfer request signal.

747. The system of claim 610, wherein said continuing is triggered by one of navigation to a triggering resource and navigation of a triggering link.

748. The system of claim 747, wherein the triggering resource is a resource belonging to a category pre-defined to cause triggering.

749. The system of claim 747, wherein the triggering resource is a resource containing a specific attribute specifying a trigger.

750. The system of claim 747, wherein the category pre-defined to cause triggering is one of a navigation level, a content type, and an activity type.

751. The system of claim 740, wherein maintenance of information needed to transfer cumulative state is routinely performed.

752. The system of claim 751, wherein said maintenance is routinely performed once support for the said device sets is first enabled at one of system setup time and device set setup time.

753. The system of claim 751, wherein said maintenance does not depend on a user activating a special state tracking activity for said session.

754. The system of claim 610, wherein said processor further performs the steps of:

having said first device set provide user identification information corresponding to a user; and

having second device set provide user identification information corresponding to said user.

755. The system of claim 754, wherein user authentication information is provided by both device sets.

756. The system of claim 610, wherein said processor further performs the step of:

transmitting from said first device set to said second device set a state record corresponding to said session as presented at said first device set.

757. The system of claim 756, wherein the state record is used at said second device set to identify a resource for browsing.

758. The system of claim 757, wherein the resource is used as a hyperlink starting resource with one of a designated set of third-party arcs to determine a hyperlink ending resource for traversal at said second device set.

759. The system of claim 756, wherein the state record comprises a current time position and a current time.

760. The system of claim 756, wherein said transmitting is triggered by a user signal.

761. The system of claim 756, wherein said transmitting is performed when a program change event occurs.

762. The system of claim 758, wherein the hyperlink ending resource is presented to a user for activation.

763. The system of claim 756, wherein said first device set initiates the transmission of said state record.

764. The system of claim 756, wherein said state record is requested from said first device set by said second device set.

765. The system of claim 610, wherein said processor further performs the steps of:

acquiring information of the identity of a resource associated with said session as presented at said first device set; and

employing said information to determine a source for enhancement resources related to said session as presented at said first device set.

766. The system of claim 765, wherein the information is obtained via a wide area network.

767. The system of claim 766, the wide area network is the Internet.

768. The system of claim 756, wherein the state record includes a navigation history, wherein said navigation history can be retraced as if said continuing had not occurred.

769. A television system head-end system, comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

receiving state information from a set-top box; and

relaying said state information over an Internet connection to a device set collocated with said set-top box.

770. The system of claim 769, wherein a subscription fee is charged for the relaying service.

771. A system for providing hyperlinks for interactive television, comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

maintaining access to a linkbase, said linkbase correlating state information with hyperlinks;

receiving a request for one or more hyperlinks, said request including interactive television state information corresponding to a particular device; and

employing said linkbase to respond to said request, wherein requests relating to any of multiple programs are submitted, said requests differing only with regard to said state information.

772. The system of claim 771, wherein the system is employed by a television service operator to offer a common portal service that supports multiple programs offered by that operator.

773. The system of claim 771, wherein said linkbase is an array.

774. The system of claim 771, wherein said linkbase is a database.

775. The system of claim 771, wherein said linkbase operates as a process.

776. The system of claim 771, wherein the interactive television state information corresponding to a particular device includes an indication of channel.

777. The system of claim 771, wherein the interactive television state information corresponding to a particular device includes an indication of broadcast time.

778. The system of claim 771, wherein the correlated state information includes indications of channels.

779. The system of claim 771, wherein the correlated state information includes indications of broadcast times.

780. The system of claim 771, wherein said request is dispatched via a computer network.

781. The system of claim 780, wherein said network is the internet.

782. A system for recording events, comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

recording in a log, at a desired level of granularity, a device's events;

identifying recorded events that have been made superfluous by changes in the device's state; and

deleting the identified events from the log.

783. The system of claim 782, wherein the system uses a model-view-controller architecture.

784. The system of claim 782, wherein the recording of events is used for a one-time process of replicating state.

785. The system of claim 784, wherein the one-time process is the initiation of a synchronization process which then continues.

786. The system of claim 785, wherein the synchronization is in support of collaboration by multiple users.

787. The system of claim 784, wherein the one-time process is the transfer of a multi-machine user interface session.

788. The system of claim 782, wherein the events are used to synchronize activity among the active members of a group of systems whose membership may vary.

789. The system of claim 788, wherein said deleting is limited to related changes that occur within synchronization intervals bounded by points in time when membership in the active group changes.

790. A system for providing interactive television services, comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

enabling a user at a television device to conduct a user session, said user session requiring services resident on an independent user computer; and

providing said device access to said services.

791. The system of claim 790, wherein said services are provided as web services using simple object access protocol.

792. The system of claim 790, wherein the computer is one of a personal computer, a personal data assistant, and a tablet

793. The system of claim 791, wherein the web services relate to one of personal productivity information and financial information.

794. The system of claim 793, wherein said personal productivity information comprises at least one of contact lists, calendars, and mail logs.

795. The system of claim 793, wherein said financial information comprises at least one of payables, receivables, stock transactions, stock portfolios, banking transactions, banking holdings, and shopping.

796. The system of claim 790, wherein said steps are conducted at one of the location of the television and a remote location.

797. The system of claim 796, wherein said remote location is a head-end.

798. A system for providing venue-based computer-related services comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

employing a venue-based device set to offer a programmatic service to a user-supplied computer collocated with said device set;

having the device set provide said service to the computer, wherein the computer's user interacts with the venue-based device set while the device set draws on the resources of the computer.

799. A system for providing a browsing session comprising:

a memory having program code stored therein; and

a processor operatively connected to said memory for carrying out instructions in accordance with said stored program code;

wherein said program code, when executed by said processor, causes said processor to perform the steps of:

beginning a session comprising browsing interactions at a first device set; and

continuing said session at a second device set.

800. A method of operating a remote media server, comprising:

receiving state information from a first device set; and

relaying said state information over an Internet connection to a second device set collocated with said first device set.

801. The method of claim 800, wherein a fee is charged for the relaying service.