A system, method and computer-readable media are disclosed for format negotiation between a graph of connected nodes. A source node for multi-media content publishes a set of format constraints associated with the multi-media data and capabilities. A destination node for presenting the multi-media data publishes its set of format constraints associated with its capabilities. At least one filter node within the graph transmits or modifies multimedia data between the source and destination nodes. The format constraints are represented by logical expressions in disjunctive normal form and describe required values or acceptable ranges of values for one or more parameters. The method aspect of the invention comprises receiving source node format data, destination node format data and at least one filter node format data. As nodes are connected to the graph of nodes, the new node’s format data is propagated to the directly and indirectly connected nodes in the graph. After the last node is connected, format negotiation resolves format constraints between the nodes in the graph. The negotiation occurs within the framework or operating system of a computing device, a remote server or a proxy server.
**FIG. 3**

```
302
COMPUTE
DEVICE
1

\[ \text{PUBLISH} \]

300

\[ \text{SERVER} \]

304

\[ \text{PUBLISH} \]

306

\[ \text{COMPUTE}
DEVICE
2 \]
```

**FIG. 4**

```
402
RECEIVING SOURCE NODE FORMAT DATA
FROM A SOURCE NODE

404
RECEIVING DESTINATION NODE FORMAT
DATA FROM A DESTINATION NODE

406
NEGOTIATING ANY UNRESOLVED FORMAT
CONSTRAINTS BETWEEN THE SOURCE
NODE FORMAT DATA AND THE
DESTINATION NODE FORMAT DATA
```
FIG. 5A

FIG. 5B
FIG. 6

REQUESTING A CONNECTION OF A NEW NODE TO THE GRAPH BY PROPAGATING A FORMAT CONSTRAINT FOR THE NEW NODE TO THE AT LEAST ONE NODE IN THE GRAPH

IF THE PROPAGATION FAILS, THEN DENYING THE REQUEST FOR CONNECTION

IF THE PROPAGATION IS SUCCESSFUL, THEN CONNECTING THE NEW NODE TO THE GRAPH
SYSTEM AND METHOD OF FORMAT NEGOTIATION IN A COMPUTING DEVICE

PRIORITY CLAIM

[0001] The present invention claims priority to U.S. Provisional Patent Application No. 60/543,108 filed on Feb. 9, 2004, and U.S. Provisional Patent Application No. 60/543,356, filed on Feb. 9, 2004, the contents of each of these cases are incorporated herein by reference.

RELATED APPLICATIONS

[0002] The present application relates to the following applications: (1) Ser. No. ______ Attorney Docket No. 4002.Palm.PSL, entitled “A Method And Graphics Subsystem For A Computing Device”; (2) Ser. No. ______ Attorney Docket No. 4003.Palm.PSL, entitled “A Method And System For A Security Model For A Computing Device”; and (3) Ser. No. ______ Attorney Docket 4004.Palm.PSL, entitled “A System And Method Of Managing Connections With An Available Network” each of which are filed on the same day as the present application; the contents of each Application are incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0003] 1. Field of the Invention

[0004] The present invention relates to a system and method of performing format negotiation to accomplish the transmission of multi-media data between a first node and a second node.

[0005] 2. Introduction

[0006] Small handheld computing devices continue to improve in their complexity and data processing capabilities. Examples of such devices include PalmOne’s Treo 600 computing device that includes capabilities for multi-media viewing and recording of pictures and video which can also include audio. These operate on various versions of the Palm Operating system. Other available devices include, for example, Hewlett Packard’s mobile devices based on the Windows PC operating system, such as the HP iPaq hx4700, the HP rx3000 or the Dell Axim X50. Such devices will each have improved hardware capabilities which may include, for example, a VGA (480x640) display and central processors, video co-processors, means for recording multi-media data and more. Some devices have been called Portable Media Centers or Mobile Media Companions since they have hardware capable of providing higher quality video and audio capabilities.

[0007] In order for video to be transmitted from a video source, such as a DVD or streaming multi-media, to the hardware display or recording destination, the source must be compatible with the capabilities of the hardware and software capabilities of the destination display/record device. This may be handled by way of multi-media standard formats, such as MPEG-2 or MPEG-4 in which the multi-media source is encoded according to the standard and the destination device/ software is compatible with that standard. An example media software application is Windows Media Player 10 Mobile (MP 10 Mobile). MP 10 Mobile is a software application running on an operating system, such as the Windows CE Operating system, that will manage digital rights management for content, display special features such as album cover art that are programmed into the source data and will receive appropriately formatted content and record the content or display the video on the display as well as producing the sound from the multi-media source.

[0008] There are limitations regarding how source content may be played using such an approach. For example, recorded TV shows may be played using MP 10 Mobile only if they were recorded with a Windows Media Center Edition PC, and then they need to be converted and transferred to the mobile computing device for viewing. MPEG and WMV video must be converted to be played on the mobile device as well. Other differences in the source and the playback device may relate to audio. A multi-media source may be recorded in stereo or in the 5.1 or 7.1 format, and the computing device may have a single speaker.

[0009] Many mobile computing devices are also periodically synchronized with a desktop PG When this synchronization occurs, multi-media content may be transmitted to the mobile device for viewing. With MP 10 Mobile, upon synchronization, the Windows Media Player 10 on the PC automatically recognizes the mobile device capabilities and adjusts its conversion settings so that the device receives video formatted for its capabilities. In this regard the software application Windows Media Play 10 on the PC running on the desktop computer’s operating system must be programmed to recognize the mobile device, its capabilities and perform the appropriate adjustments to match the mobile device capabilities.

[0010] When the software that manages the display of video is running separate from the operating system of the computing device as in the MP 10 Mobile software, challenges will exist for third party software developers that develop software containing multimedia components. Third party software for video games or that may include video clips must be developed to anticipate the capabilities of the destination computing device. For example, third party software may include MPEG encoded video. Given the variety of hardware devices and software applications such as MP 10 Mobile that are in the marketplace, the need for software developers to insure that their software will be compatible and operable on a variety of devices increases the difficulty, complexity and cost of developing software. Such complexities can further inhibit or slow down the sales and success of both the third party software program and the sales of the computing devices on which they are designed to run.

[0011] What is needed in the art is a simplified method for insuring that source capabilities and destination capabilities are compatible and that appropriate configuration is accomplished to establish a media stream between the source and destination.

SUMMARY OF THE INVENTION

[0012] Additional features and advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent
from the following description and appended claims, or may be learned by the practice of the invention as set forth herein.

The present invention addresses the needs in the prior art for an improved system and method of managing the differences in source multimedia content and destination display/recording hardware and software capabilities. The present invention comprises a system, method and computer-readable media that perform format negotiation of parameters or format constraints associated with a source node and format constraint associated with a destination node. The format negotiation occurs within a graph consisting of two or more nodes.

The method aspect of the invention relates to establishing a multimedia format for communicating data between a source node and the destination node. The invention relates to any transformation of media data whether the data is in a raw form such as RGB or YUV video frame or encoded media data such as MPEG1 or MPEG4, for example. The source node may be, for example, a third party software program on a computing device which may manage the reception of images or video through a camera on the device or attached to the device. An example of the destination node is the computing device display or means for recording data and software that controls the presentation of data on the display. The method comprises receiving source node format data from a source node, receiving destination node format data and negotiating any unresolved format constraints between the source node format data and the destination node format data. In addition to the source node and destination node, a graph of nodes may consist of any number of nodes between the source and destination nodes. In this context, the method comprises propagating format constraints for each node to directly or indirectly connected nodes in order to accomplish format negotiation. The communication of multimedia data (which may be in the form of raw data, encoded data or any other form of multi-media data) may be in the context of playback and/or recording of data.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

- FIG. 1 illustrates the basic components of a computing device according to the present invention;
- FIG. 2 illustrates an example computing device and format negotiation;
- FIG. 3 illustrates a server or proxy server aspect of the invention;
- FIG. 4 illustrates a method embodiment of the invention;
- FIG. 5A illustrates the use of media nodes and endpoints;
- FIG. 5B illustrates a graphical aspect of the invention; and
- FIG. 6 illustrates another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Various embodiments of the invention are discussed in detail below. While specific implementations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without parting from the spirit and scope of the invention.

The present invention provides for a system, method and computer-readable media that perform format negotiation between a source node for multi-media information and a destination node for playback or recording of multi-media information. In a more general statement, the invention provides for format negotiation between any two nodes that publish their possible multimedia formats within a system. A graph of nodes comprises a first source node, one or more intermediate nodes or filter nodes and a second destination node. In general, the intermediate nodes are nodes within the graph between the source and destination nodes that have responsibilities such as transforming or converting data from one format to another format. There may also be cascading format negotiations along a communication path from node to node or a propagation of a format from node to node as will be described more fully below. The graph is a representation of the interconnected nodes through which media data will pass through the system. The nodes consume, modify or produce media. Any node may have zero or more inputs and zero or more outputs and will have at least one connection to modules or other nodes outside the current node. An example multi-media node is one that transmits audio and/or video data. Inasmuch as one embodiment of the invention relates to a hardware device or system, the basic components associated with a computing device are discussed first.

FIG. 1 and the related discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described, at least in part, in the general context of computer-executable instructions, such as program modules, being executed by a personal computer or handheld computing device. Generally, program modules include routine programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.
[0026] With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device 100, including a central processing unit (CPU) 120, a system memory 130, and a system bus 110 that couples various system components including the system memory 130 to the CPU 120. The system bus 110 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory 130 includes read only memory (ROM) 140 and random access memory (RAM) 150. A basic input/output (BIOS) contains the basic routine that helps to transfer information between elements within the personal computer 100, such as during start-up, is stored in ROM 140. The computing device 100 further includes a storage device such as a hard disk drive 160 for reading from and writing data. This storage device may be a magnetic disk drive for reading from or writing to removable magnetic disk or an optical disk drive for reading from or writing to a removable optical disk such as a CD ROM or other optical media. The drives and the associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules and other data for the computing device 100.

[0027] Although the exemplary environment described herein employs the hard disk, the removable magnetic disk and the removable optical disk, it should be appreciated by those skilled in the art that other types of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memory (RAM), read only memory (ROM), and the like, may also be used in the exemplary operating environment.

[0028] FIG. 1 also shows an input device 160 and an output device 170 communicating with the bus 110. The input device 160 operates as a source for multi-media data or other data and the output device 170 comprises a display, speakers or a combination of components as a destination for multi-media data. The device 170 may also represent a recording device that receives and records data from a source device 160 such as a video camcorder. A communications interface 180 may also provide communication means with the computing device 100.

[0029] As can be appreciated, the above description of hardware components is only provided as illustrative. For example, the basic components may differ between a desktop computer and a handheld or portable computing device. Those of skill in the art would understand how to modify or adjust the basic hardware components based on the particular hardware device (or group of networked computing devices) upon which the present invention is practiced.

[0030] As shown in FIG. 2, the present invention relates to displaying or recording multi-media information on a computing device 200. The computing device 200 comprises a node that may be referred to as a destination node 202 such as a display node communicating with the node 202, for example. The node 202 will often have certain format constraints associated with it. For example, the node 202 may be the software and hardware capability to display both YUV and RGB formatted video. The YUV format is a color encoding scheme for natural pictures in which the luminance and the chrominance information are separate. The human eye is less sensitive to color variations than intensity variations. Thus, YUV allows the encoding of luminance (Y) information at full bandwidth and chrominance (UV) information at half bandwidth. The RGB format (Red, Green, Blue) are the three primary colors used in video processing, often referring to the three un-encoded outputs of a color camera. The RGB color model is based on the mixing of red, green and blue. The combination and intensities of these three colors can represent the whole spectrum A destination node may be any destination device such as a file, network stream, a screen and speakers.

[0031] One aspect of the invention is that the destination node 202 simply publishes 208 its format constraints to achieve a communication link to another node or nodes for transmission or reception 212 of multimedia data. Each node within the graph will have certain constraints on what input media it accepts or can operate on and also publishes the format associated with the node output. A published format may be a logical expression represented by a vector, an object, a set of constraints or a set of values as format constraints. In one example, a format parameter may be a component of a media library and represents a multimedia format. They may be associated with several objects in a multimedia subsystem, such as media endpoints and codec objects (extractors, decoders, encoders and composers) to describe the data formats that a node can read or write. The format constraints may be packaged together into a vector object. The published format preferably comprises a format type. The following format constraints associated with identifying a format type within a header file are provided by way of example. In the framework, they may be declared in a multi-media format definition header file to be grouped into distinct types, such as MPEG4, raw audio, raw video, MIDI audio and so forth. FIG. 2 shows the second node or destination node 202 publishing 208 its set of format constraints to the “black box” 204. The box 204 may be a framework associated with the operating system of a computing device, a graph of intermediate communication nodes or any other intermediate structure. The benefit shown in FIG. 2 is that these nodes do not require knowledge of the capabilities of other nodes in the system but merely need to publish their format constraints to achieve communication of multi-media data.

[0032] Another example criterion for the display/record node is that the display/recording of YUV is hardware accelerated which includes a constraint that the width of the video must be a multiple of 16. Wildcard parameters may also be published or the node may not publish any value for a particular parameter which indicates that any value is acceptable.

[0033] A source node 206 also publishes 210 its format constraints. An example source node 206 is a video decoder that outputs either YUV and RGB formats or a video camcorder producing raw audio/video data. Such decoders may be, for example, an MPEG decoder. Other format constraints associated with the display/record node 202 and source node 206 may also be available. For example, the video decoder may publish the following set of format constraints: (width=160 AND height=120 AND colorspace=RGB) OR (width=160 AND height=120 AND colorspace=YUV). This parameter set indicates that the width must be 160, the height must be 120 and the colorspace is flexible and may be either RGB or YUV. When a movie or video content is played or recorded, the source node 206 and
destination node 202 must have some compatibility to achieve transmission or reception 214 of the multi-media data. According to an aspect of the present invention, the operating system (preferably) manages format negotiation to achieve this compatibility. The manner in which the nodes publish their format constraints enables the system or some other entity, including the nodes themselves, to negotiate within the constraints. The display/record node 202 publishes, for example: (colorspace=RGB, height=[240, 120, 60]) OR (colorspace=YUV AND width is multiple-of 16) to the operating system 204. Thus indicating that the width and height may be any value, the colorspace must be either RGB or YUV and if it is YUV, the width must be a multiple of 16. The Boolean algebra guarantees that any given Boolean term can always be reduced or transformed to an equivalent term in normal-form, without the disjunctive normal form (DNF) or the conjunctive normal form (CNF). For example, if A, B, C, D, E are constraints associated with a node, i.e., A="width"<100; B="height">20 and so forth, then the DNF form is: (A AND B) OR (C AND D AND E) (other combinations of course may be used as well) and the CNF form is: (A OR B) AND (C OR D OR E) (or in other combinations. The two forms shown above are not equivalent but serve to illustrate the nature of the normal forms. The preferred embodiment of the invention utilizes the DNF form for the media-format. A DNF form example would be: ("width"<100 AND "height">200) OR ("width"<200 AND "height"<400). The example describes the media-format with two alternatives.

[0034] If a node with an output is connected to a node with an input in the process of generating a graph of nodes, there are both output constraints as well as input constraints. In order to form a connection, it is necessary to negotiate the two sets of constraints (if they can be negotiated at all), i.e.: C1 ("width" in {100,200}) AND ("height" in {50,60}), C2 ("width"<150) AND ("height"<60) can be negotiated by simply AND'ing the boolean terms. In another example, the constraints C1 AND C2 =>("width"<100) AND ("height"<50) produce a single fully resolved alternative of width=100 and height=50 to allow the communication of media data between the two nodes.

[0035] Format negotiation can fail in several ways: either the constraints are not restrictive enough so that even after negotiation there are variables to choose from or if there is no way to satisfy both constraints simultaneously. If both constraints cannot simultaneously be satisfied, it is not possible to connect the two nodes. The following is an example of two published constraints that are impossible to negotiate: C1: "width"<400, C2: "width"<50. The following shows an example of two constraints with variables left after reduction: C1: "width"<100, C2: "width"<50. In this case, any width less than 50 will do, but some higher authority still needs to actually choose a concrete value for width satisfying this constraint.

[0036] The examples above typically apply to format negotiation between two nodes. However, as will be discussed below, if the information of the entire graph or at least a significant portion of it is known ahead of time, more sophisticated format negotiation can be performed.

[0037] The source node publishes its information and the operating system, framework or other module such as a remote server or proxy server receives all the published data and performs format negotiation to achieve a connection between the source and the destination nodes. The constraints are preferably a well-known name or value such as "colorspace", a relationship such as equals or less than, and a constant value. In the preferred embodiment of the invention, no relationship between the names is allowed in the format constraint publication.

[0038] A strength of the invention is that a format constraint published by a node may contain any number of parameters as defined by the node and not the framework. The framework itself does not attempt to describe each parameter that may be associated with a constraint. In this manner, because the framework has a purpose of resolving format constraints instead of defining a particular set of restraints, the invention enables an expandable capability for communication of media data between any number of nodes connected in a graph.

[0039] Given the above example, the operating system 204 analyzes the published format constraints for the source node 206 and destination node 202 and produces the resulting negotiated format as: (width=160 AND height=120 AND colorspace=YUV). In another example, assume that a video having a 120x90 size is to be played. With that size and width ratio, the resulting negotiated format would be: (width=120 AND height=90 AND colorspace=RGB) because 120 is not a multiple of 16, and so the YUV video overlay cannot be used. Therefore, the system is forced to use the RGB colorspace.

[0040] As mentioned above, it is preferable in the present invention that the operating system or basic framework and not the nodes or separate software application such as a MP 10 Mobile that decides which format to use. One benefit of this approach is that third party software developers only need to publish the format constraints and source node data when their software will run on a framework that performs format negotiation. This greatly simplifies the coding process for multi-media applications and reduces development time and costs. With the published format constraints, the framework negotiates a compromise format for the playback of the multi-media information. As mentioned above, the framework that actually performs the format negotiation is preferably the operating system but may be any separate module (hardware or software) and may be performed remotely or on a proxy server.

[0041] There are many variations on how the format constraints may be published by the nodes. In addition to the examples set forth above regarding the form of the constraints, the format constraints may be more complicated logical expressions as well. In this regard, the format constraints are preferably represented in a disjunctive normal form such as:

[0042] (key1<relation=value1) AND (key2<relation=value2) AND . . . ) OR ((key1<relation=value3) AND . . .)

[0043] In other words, a logical expression may be a format constraint that is a set of terms conjoined by AND and OR. In this context, a format constraint may comprise an AND'd set of terms, wherein a format vector comprises an OR'd set of format constraints or format alternatives. A parameter in a format constraint may be related to a value or a range of values but are preferably independent of other
parameters. However, format alternatives may be used to get an equivalent effect in many cases. For example, if “x” and “y” may take the value of 1 or 2, but may not be equal, this can be expressed as follows: ((x=1) AND (y=2)) OR ((x=2) AND (y=1)).

[0044] With this flexibility, nodes may publish not only logical expressions but requirements, preferences, alternative values, wireless-related values, quality of service, power consumption, pricing plans and any other variations. The algorithms associated with the framework can therefore maximize the resulting set of format constraints for displaying or recording the data based on the ability to resolve the constraints in the parameter sets and to maximize the ultimate transmission of data from the source node to the destination node. As mentioned above, in previous systems, the software applications where responsible for doing the negotiation and capability analysis. In the present invention, the “nodes” whether they are a software application, display device, recording device and so forth, only need to publish their format constraints and the framework manages all the negotiation.

[0045] With regards to format preferences, a developer may create a format preference object with variables such as a key, a value and a weight. The key specifies an attribute of a multimedia object (such as bit rates for MPEG and successor formats, number of frames processed per second, number of frames per buffer for audio or video data, width in native pixels of a video frame or graphics file, height of a video frame in native pixels of a video frame or graphics file, audio decoder parameters etc.). The value specifies the preferred value for that attribute and the weight specifies how much that value is preferred. The weight may only be used if two competing preferences are specified. In that case, the one of the greater weight is used.

[0046] An example of using a format preference object is provided next. A format preference object is created within a media endpoint object to specify a format preference. Suppose a developer has a media node that can take any byte order but works more efficiently if the byte order is a little endian. This node’s endpoints would have a format that either did not specify a format term for the P-FORMAT-KEY_BYTE_ORDER attribute or had one with a wild value. To specify that the little-endian order is preferred, it should pass a format preference object that uses the P-FORMAT-KEY_BYTE_ORDER key with a value of little endian. After creating the appropriate format constraints to publish from a node or an object, the developer does not interact with the system any further with regards to the formats. The format, format vector and endpoint correctly handle all format comparisons via the framework at the appropriate times.

[0047] The framework on the computing device preferably performs the negotiation, but other devices such as a separate server or a proxy server which selects alternate content or transforms content format to optimize it for the target device. This aspect of the invention is shown in FIG. 3. A first compute device 302 is the display/record device in this example and publishes its format constraints for display or recording. A source device 304 receives its format constraints. A server 306 remote from both the compute devices 302, 304 receives the published format constraints and negotiations the ultimate formatting information for the transfer of the multi-media data from the source device 304 to the destination device 302. Once the format constraints are negotiated, the server 306 may then be the conduit for the transmission of the data or there may be another communication path via the Internet, a wireless network, or some other communication path for the data. Furthermore, the server 306 may also process the source data is an optimization could occur to improve the quality or service, use of bandwidth, pricing or some other parameter. The optimization may occur in a transformation of the multi-media data in some manner to match a preferred setting of one of the nodes or may comprise a modification of a communication path to increase the bandwidth of the path or improve a parameter of the transmission of the multimedia data.

[0048] In one aspect of the invention, the format negotiation may involve testing to see if the framework can change a parameter associated with a node based on a certain criteria. For example, if significant improvement may be gained if a particular parameter were changed for the source node, the negotiation may test whether the source node would accept that alteration even if the alteration is outside of the original published set of format constraints. The framework may also manage timing and buffering of data.

[0049] Software developers may be able to prepare software applications using a multimedia library comprising a multimedia API that enables them to access call services associated with the operating system. The library provides public SDK-level APIs that multimedia clients use to access multimedia features.

[0050] FIG. 4 illustrates a method embodiment of the invention. The method may be practiced upon a computing device, a server, a proxy server and so forth. There is no restriction on the hardware configuration on which the method is practiced. The method for establishing a multimedia format for communicating data between a source node and the destination node comprises receiving source node format data from a source node (402), identifying a destination node having destination node format data (404) and negotiating any unresolved format constraints between the source node format data and the destination node format data (406). The destination node may be previously identified or selected in which case the method merely receives the published data of the destination node for the negotiation process. As mentioned above, as part of this basic process there may be alternate steps that are not essential to the invention and may or may not be practiced. For example, at least one of the source node format data and the destination node format data may comprise: a parameter with a range of acceptable values, a required value and a preferred value or range of values or logical expression.

[0051] The method may further comprise, based on the source node format data and the destination node format data, selecting a communication node for communicating the multimedia data between the source of node and the destination node. The black box 204 may be, for example, a media encoder or decoder (codec) component. Where intermediate nodes are in the communication path between a first node and a second node, cascading format negotiation may occur in between sets of nodes throughout the communication path.
In one scenario where a communication node or communication nodes are used to connect a source node and a destination node, the method may comprise generating a graph of connections between source nodes and destination nodes. For example, a source device and destination device may publish their device requirements, these format constraints may be used by the framework to determine which communication nodes and which nodes to use for connecting the source multimedia data to the destination device for playback.

Several characteristics of communication nodes (or media nodes) are discussed next. A communication node may have the responsibility for obtaining and transmitting buffers. A communication node has one or more endpoints that are portals through which a media node sends buffers to or receives buffers from another media node. Endpoints are either inputs or outputs. By connecting media node endpoints together, one can construct a graph of media nodes. The endpoints can also publish the list of media format constraints that they particular media node works with for format negotiation.

FIG. 5A illustrates the use of media nodes 520, 522, 524, 526, media endpoints 530, 532, 534, 536, 538, 540 and buffers. The endpoints obtain buffers from a buffer source node 542. As can be seen, some media nodes 522 can have one input endpoint 532 and multiple output endpoints 534, 536. A media node is not required to have both input and output. If a media node receives its data from output the framework, it has only an output. The media node packages data that data in a buffer and sends it to another media node on the output. Such a node is called a producer node because it produces buffers. Similarly, a media node that receives a buffer from a media node and sends its data to something outside the framework has only an input and is called a consumer node. Media nodes with both an input and an output are called filter nodes. Filter nodes may process the buffer in some way, or they may just pass it on unchanged.

An example graph is shown in FIG. 5B which may provide the user or an administrator with a visual image of the connection between the source and the destination nodes. FIG. 5B illustrates a screen 500 illustrating a graphical path from the source node 502 through a filter node 1504, a filter node 2506 to the destination node 508. The graph may or may not be available for viewing by a user.

If a graph is generated of the negotiated connection, the method may comprise negotiating the unresolved format constraints according to the generated graph. In one aspect of the invention, the step of negotiating the unresolved format constraints comprises selecting one of the range of acceptable values for the communication of data. If both the source node has a range of acceptable values for a parameter and the destination node also includes a range of values for the same parameter (such as width), then format negotiation may comprise selecting a first value for the source node from the range of acceptable values that matches a second value from the range of acceptable values for the destination node. For example, if a first node publishes a parameter “A=15” and the second node publishes “A=10” then there is an overlapping range of compatible values from 11-14 that format negotiation will resolve to select the appropriate value for within the compatible range. Furthermore, optimization and maximization of a value such as quality of service, predefined preferences, bandwidth or pricing may further be employed to select the negotiated value. For example, if a predefined preference based on some criteria establishes that A is preferably within the range of 8-11, then the format negotiation would select a value of A based on the compatible range 11-14 and the known preference.

FIG. 5B illustrates a first filter node 504 and a second filter node 506. The path represented from the source node 502 to the destination node 508 passes through multiple intermediate nodes. One aspect of the invention comprises multiple format negotiations between the various nodes in the path. Given the differences in bandwidth, licensing, hardware, and so forth that may exist in between any two nodes (whether it be within a single computing device or over a diverse network), many format negotiations may need to occur to achieve optimal transmission of the data.

In another aspect of the invention, negotiating unresolved format constraints further comprises identifying alternate contents or alternately transforming content formats to optimize content for the destination node. As with the other aspects of the invention, these steps may be practiced on the framework, a remote server or proxy server used to identify the alternate content or alternate transform content formats.

In yet another aspect of the invention, shown in FIG. 5B, node 504 splits the multi-media signal and transmits it to two nodes, node 506 and filter node 3510. An example of when this may be done is if the source node 502 wants to transmit an MPEG file for a destination node, the filter node 504 may split the MPEG (or other protocol) data into an audio stream and a video stream. Each of the streams is then transmitted to a separate node 510 or 506 for decoding the audio/video and transforming it into another format. Raw data is then transmitted from these nodes 510, 506 to the destination node 508 for audio and video presentation.

Selection of which filter nodes to use may comprise several steps, such as (1) identifying the source data format (such as MPEG) and finding the correct nodes for further processing, and (2) receiving the specification of what nodes support which formats and analyzing other format constraints associated with each node. In this regard, once multiple nodes have been identified, then multiple format negotiations between any two nodes will occur to complete the connection between the source node and the destination node.

An example will further illustrate multiple format negotiations and also data transformation based on the format negotiation. Assume the source node 502 will output audio data and publishes the appropriate format constraints. Format negotiation may or may not take place between the source node 502 and the first filter node 504. The node 504 would publish that it will be outputting MPEG encoded audio data and the system would match node 504 with a decoder node 506 that can handle (decode) the particular MPEG data. Assume that destination node 508 is a single speake. The node 506 then would publish to the destination node 508 that it will be outputting raw PCM data in, for example, two channels for stereo sound. Its output format constraints such as assemble rate of the audio are example
format constraints. Alterations to the data may occur as part of the negotiation. If the destination node 508 is a single speaker, and the PCM raw output from decoder node 506 is two speaker channels, based on the format negotiation process which identifies that a parameter of speaker node 508 is it is a single channel, the stereo channels may be downmixed into a single channel before transmission to the destination node 508. In this manner, the format negotiation enables the appropriate data transformation to match the data with the destination.

[0062] FIG. 6 illustrates another embodiment of the invention. This embodiment relates to the propagation of format constraints through a graph of nodes. A graph of nodes comprises plurality of nodes and may involve more than two nodes. As a graph of nodes is generated as is shown in FIG. 5B, new nodes are connected to the graph. This aspect of the invention involves determining when a new node may be connected and how the final resolution of the format constraints for each node is completed. Format propagation is the process of adding nodes to a graph by propagating the format constraints for a new node through each node or node endpoint in a graph to determine whether the format constraint for the new node are compatible with the format constraints of the nodes in the graph. In this regard, when an endpoint is attached to a graph, its format constraint is propagated to all the endpoints to which it is directly or indirectly connected. When the concept of a node being “connected” to a graph, a media endpoint as shown in FIG. 5A is preferably the component that is used to publish format constraints for that node and that is connected to the graph.

[0063] FIG. 6 illustrates the steps of the method. As a graph is built, it starts with one node with new nodes requesting connection to the graph. The method comprises requesting a connection of a new node to the graph by propagating a format constraint for the new node to the at least one node in the graph (602). The format constraint is propagated through the endpoints in the graph which are directly or indirectly connected to the attached endpoint. The new node publishes its format constraint and the system compares the new node format constraints with the format constraints in each node already within the graph to determine compatibility of format constraints. The system preferably does not resolve format constraints at this stage. If the propagation fails, then the method comprises denying the request for connection (604). If the propagation is successful, meaning, for example, the format constraints for the new node are identical as compatible with the nodes already connected to the graph, then the method comprises connecting the new node to the graph (606).

[0064] The steps (602)-(606) are repeated for each new node that requests connection to the graph. This process continues until the source node, all intervening media or connection nodes, and a destination node are connected in a graph with compatible format constraints. After each new node is connected to the graph, the method further comprises resolving all format constraints for each node in the graph. This approach enables in a single atomic transaction the full resolution of all the format constraints across the graph for each node or each node media endpoint.

[0065] The task to be performed by any given node in a graph is in material to the present invention. The propagation and resolution of format constraints within a graph is indifferent to whether a node is a codec, a filter, a recording device, an abstraction of hardware on the computing device such as a video/audio input or output.

[0066] Embodiments within the scope of the present invention may also include computer-readable media for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code means in the form of computer-executable instructions or data structures. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or combination thereof) to a computer, the computer properly views the connection as a computer-readable medium. Thus, any such connection is properly termed a computer-readable medium. Combinations of the above should also be included within the scope of the computer-readable media.

[0067] Computer-executable instructions include, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. Computer-executable instructions also include program modules that are executed by computers in standalone or network environments. Generally, program modules include routines, programs, objects, components, and data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represents examples of corresponding acts for implementing the functions described in such steps.

[0068] Those of skill in the art will appreciate that other embodiments of the invention may be practiced in network computing environments with many types of computer system configurations, including personal computers, handheld devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. Embodiments may also be practiced in distributed computing environments where tasks are performed by local and remote processing devices that are linked (either by hardwired links, wireless links, or by a combination thereof) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0069] Although the above description may contain specific details, they should not be construed as limiting the claims in anyway. Other configurations of the described embodiments of the invention are part of the scope of this invention. For example, the format negotiation may exist on a single compute device and between nodes within that device or the negotiation may occur over a communications
network between nodes of the network. The format constraints may take the form of equations having various parameters and variables. Accordingly, the appended claims and their legal equivalents should only define the invention, rather than any specific examples given.

We claim:

1. A method for establishing a multimedia format for communicating data between a first node and the second node, the method comprising:
   - receiving first node format constraint data;
   - receiving second node format constraint data; and
   - negotiating any unresolved format constraints between the first node format constraint data and the second node format constraint data.

2. The method of claim 1, wherein at least one of the first node format constraint data and the second node format constraint data comprises at least one parameter comprising a range of acceptable values.

3. The method of claim 2, wherein negotiating any unresolved format constraints further comprises identifying a parameter within the range of acceptable values.

4. The method of claim 1, further comprising:
   - based on the first node format constraint data and the second node format constraint data, selecting a filter node for communicating the data between the first node and the second node.

5. The method of claim 4, wherein the filter node transforms the data.

6. The method of claim 1, further comprising generating a graph of at least one connection between the first node and the second node.

7. The method of claim 6, wherein the generated graph further comprises at least one filter node connected between the first node and the second node.

8. The method of claim 6, wherein the negotiation of unresolved format constraints occurs according to the generated graph.

9. The method of claim 6, wherein the step of negotiating unresolved format constraints occurs for all nodes in the graph in a single transaction after all nodes have been connected to the graph.

10. The method of claim 1, wherein a framework that communicates with the first node and the second node performs the steps of receiving the first node format constraint data, receiving the second node format constraint data and negotiating any unresolved format constraints.

11. The method of claim 10, wherein negotiating the unresolved format constraints comprises selecting one of the range of compatible values according to the published format constraints of the first node and second node for the communication of data.

12. The method of claim 11, wherein negotiating the unresolved format constraints comprises selecting a first value for the first node from the range of acceptable values that is compatible with a second value from the range of acceptable values for the second node.

13. The method of claim 12, wherein negotiating any unresolved format constraints comprises selecting a set of format data that is compatible with the received first node format constraint data, the second node format constraint data any logical expressions associated with the first node format constraint data or the second node format constraint data.

14. The method of claim 1, wherein the format constraint data comprises at least one logical expression in disjunctive normal form.

15. The method of claim 1, wherein negotiating any unresolved format constraints further comprises identifying alternate contents or an alternate transform content formats to optimize content for the second node.

16. The method of claim 15, wherein a remote server or proxy server is used to identify the alternate content or alternate transform content formats.

17. The method of claim 1, wherein a proxy server performs the step of negotiating remove from a computing device associated with the first node and the second node.

18. The method of claim 1, wherein the first node format data comprises at least one wild card parameter.

19. The method of claim 18, wherein the second node format constraint data comprises at least one wild card parameter, and wherein negotiating any unresolved format constraints comprises selecting any value for the wild-card parameter.

20. The method of claim 1, wherein the first node is a source node and the second node is a destination node.

21. The method of claim 1, wherein the first node is a source node and the second node is a filter node.

22. A system that establishes a multimedia format for communicating data between a first node and a second node, the system comprising:
   - a module configured to receive a first node format constraint data;
   - a module configured to receive a second node format constraint data; and
   - a module configured to negotiate any unresolved format constraints between the first node format constraint data and the second node format constraint data.

23. A system that establishes a multimedia format for communicating data between a first node and a second node, the system comprising:
   - means for receiving a first node format constraint data;
   - means for receiving second node format constraint data; and
   - means for negotiating any unresolved format constraints between the first node format constraint data and the second node format constraint data.

24. A computer readable medium storing instructions for controlling a computing device to establish a format for communicating data between a first node and a second node, the instructions comprising:
   - receiving first node format constraint data;
   - identifying a second node having a second node format constraint data; and
   - negotiating any unresolved format constraints between the first node format constraint data and the second node format constraint data.
25. A method of generating a graph for use in communicating data from a source node to a destination node, the graph comprising at least one node, the method comprising:

(1) requesting a connection of a new node to the graph by propagating a format constraint for the new node to the at least one node in the graph;

(2) if the propagation fails, then denying the request for connection; and

(3) if the propagation is successful, then connecting the new node to the graph.

26. The method of claim 25, wherein a successful propagation further comprises determining whether the format constraint for the new node is compatible with format constraints for the at least one node in the graph.

27. The method of claim 25, further comprising repeating steps (1)-(3) for each new node to be connected to the graph.

28. The method of claim 27, wherein after each new node is connected to the graph, the method further comprises:

resolving all format constraints for each node in the graph.

29. A system for communicating data from a source node to a destination node using a graph, the graph comprising at least one node, the system comprising:

(1) means for requesting a connection of a new node to the graph by propagating a format constraint for the new node to the at least one node in the graph;

(2) means for denying the request for connection if the propagation fails; and

(3) means for connecting the new node to the graph if the propagation is successful.

* * * * *