



(22) Date de dépôt/Filing Date: 2001/12/19

(41) Mise à la disp. pub./Open to Public Insp.: 2003/06/19

(51) Cl.Int.⁷/Int.Cl.⁷ G06F 17/00, G06F 17/60, G06F 3/14

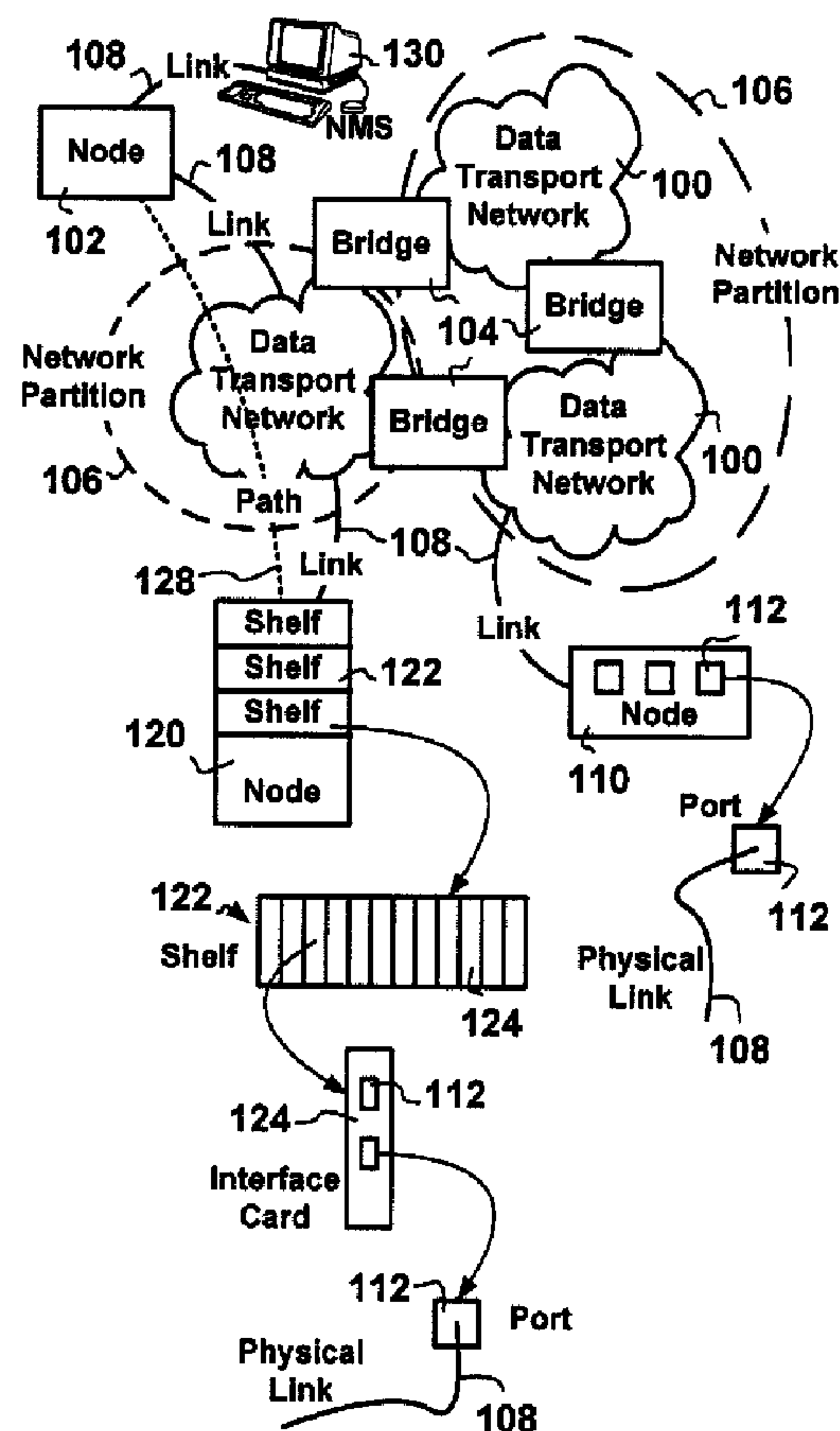
(71) Demandeur/Applicant:
ALCATEL CANADA INC., CA

(72) Inventeurs/Inventors:
PROULX, LORRAINE, CA;
TRUDEAU, DOMINIQUE, CA;
NIJEMCEVIC, SASA, CA

(74) Agent: MARKS & CLERK

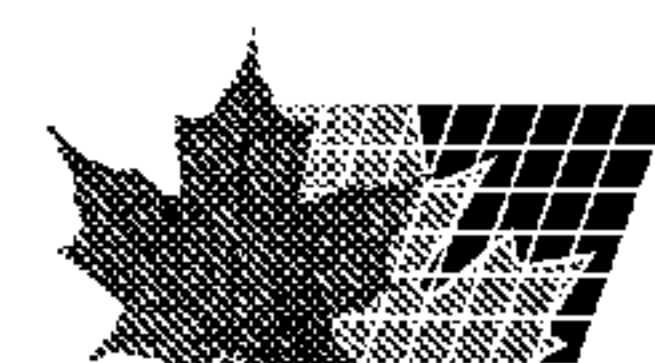
(54) Titre : INTERFACE HUMAIN-MACHINE A CONFIGURATION DYNAMIQUE

(54) Title: DYNAMICALLY CONFIGURABLE HUMAN-MACHINE INTERFACE



(57) Abrégé/Abstract:

A framework is presented to provide support for dynamically configuring a human-machine interface in a network management and service provisioning environment. The framework implements a file parser, methods of accessing a managed object server



(57) Abrégé(suite)/Abstract(continued):

to interact with instances of managed data network entity objects, a plurality of view panel components and a command interpreter. A plurality of run-time loadable form files are provided to specify the functionality, look-and-feel of view panel instances used by an analyst manipulate information in interacting with the network management and service provisioning environment. The view panel components are generically coded to provide support for the dynamically configured human-machine interface. The parser is used to load the form files and instantiate view panels on demand to created listing, filtering, creation as well as configuration forms. Advantages are derived from a separation between network management and service provisioning software application coded functionality and the specification of the interface presented to the analyst. The separation also provides for an efficient development, maintenance and deployment of the network management and service provisioning solution.

Abstract

A framework is presented to provide support for dynamically configuring a human-machine interface in a network management and service provisioning environment. The framework implements a file parser, methods of accessing a managed object server to interact with instances of managed data network entity objects, a plurality of view panel components and a command interpreter. A plurality of run-time loadable .form files are provided to specify the functionality, look-and-feel of view panel instances used by an analyst manipulate information in interacting with the network management and service provisioning environment. The view panel components are generically coded to provide support for the dynamically configured human-machine interface. The parser is used to load the .form files and instantiate view panels on demand to created listing, filtering, creation as well as configuration forms. Advantages are derived from a separation between network management and service provisioning software application coded functionality and the specification of the interface presented to the analyst. The separation also provides for an efficient development, maintenance and deployment of the network management and service provisioning solution.

M&C file No.

Client/Assoc. file No.

Dynamically Configurable Human-Machine Interface

Field of the invention

[01] The invention relates to network management and service provisioning, and in particular to methods of providing a dynamically configurable network management and service provisioning solution.

Background of the invention

[02] In the field of data network management, data transport networks are made up of a collection of managed data transport equipment. Data services are provisioned over the managed data transport equipment.

10 [03] In a competitive market place, due to a recent explosive technological development, the network management and service provisioning task is complicated by many factors including: multiple data network equipment vendors having multiple approaches in implementing the data network equipment; a multitude of data transport technologies, with each vendor
15 specializing in a sub-group of the multitude of data transport technologies; a multitude of network management and service provisioning protocols, with each vendor implementing only a sub-group of the multitude of network management and service provisioning protocols; a multitude of auxiliary network management and service provisioning equipment employing yet
20 another multitude of network management and service provisioning technologies; etc.

[04] Data transport equipment includes, but is not limited to: data switching equipment, routers, bridges, access nodes providing a multiplexing function, Remote Access Servers (RAS), distribution nodes providing a demultiplexing

M&C file No.

Client/Assoc. file No.

function, Customer Premise Equipment (CPE), etc. with next generation data transport equipment in development.

[05] With regards to data network equipment, for example data switching nodes schematically shown in FIG. 1, a vendor may chose to implement an integral device 110 having a data switching processor and a group of ports 112, while another vendor may chose a customizable implementation of a data switching node 120 including: a switching fabric, an equipment rack divided into shelves, each shelf 122 having slot connectors for connection with interface cards, each interface card 124 having at least one port 112. Although conceptually the two the data switching nodes 110 and 120 provide the same data switching function, each implementation is adapted for a different environment: the former data switching node 110 is more adapted to enterprise solutions as a private data network node, perhaps further adapted to enable access to public data services; while the latter data switching node 120 is better adapted for high data throughput in the core of public data transport networks. Typically the former (110) implements a small number of data transport protocols while for the latter (120), data transport protocols are implemented on interface cards 124 and/or ports 112 – providing for a flexible deployment thereof. All data network equipment is subject to design choices which are bound to be different from vendor to vendor.

[06] Data transport technologies include: electrical transmission of data via copper pairs, coaxial cable, etc: optical transmission of data via optical cables; free space optical interconnects, etc.; wireless transmission of data via radio modems, microwave links, wireless Local Area Networking (LAN), etc.; with next generation data transport technologies under development.

[07] Data transport protocols used to convey data between data transport equipment includes: Internet Protocol (IP), Ethernet technologies, Token-Ring technologies, Fiber Distributed Data Interface (FDDI), Asynchronous Transmission Mode (ATM), Synchronous Optical NETwork (SONET)

M&C file No.

Client/Assoc. file No.

transmission protocol, Frame Relay (FR), X-25, Time Division Multiplexing (TDM) transmission protocol, Packet-Over-SONET (POS), Multi-Protocol Label Switching (MPLS), etc. with next generation data transport protocols in development.

5 [08] The physical data network equipment alluded to above is part of larger body of managed data network entities enabling the provision of data services. The data network entities also include, but are not limited to: virtual routers, logical ports, logical interfaces, end-to-end data links, paths, virtual circuits, virtual paths, etc.

10 [09] Network management and service provisioning enabling technologies include, but are not limited to protocols: Simple Network Management Protocol (SNMP), Common Management Information Protocol (CMIP) etc.; as well as devices: special function servers, centralized databases, distributed databases, relational databases, directories, network management systems, etc.
15 with next generation devices and technologies under development.

[10] Network management and service provisioning solutions include Network Management Systems (NMS) 130 enabled via special purpose software applications coded to configure and control the above mentioned data network entities. Such software applications include, but are not limited to:
20 inventory reporting, configuration management, statistics gathering, performance reporting, fault management, network surveillance, service provisioning, billing & accounting, security enforcement, etc.

[11] It is a daunting task to provide network management and service provisioning solutions taking into account the permutations and combinations
25 of the elements presented above. Prior art approaches to providing network management and service provisioning solutions include the coding of hundreds of software applications with knowledge of hundreds of data networking entities using tens of data transmission and network management

M&C file No.

Client/Assoc. file No.

protocols. Some prior art solutions attempt to code all-encompassing large monolithic network management and service provisioning software applications.

5 [12] Coding, deploying, maintaining, and extending such software applications for network management and service provisioning has been and continues to be an enormous undertaking as well as an extremely complex procedure. Such software applications require a large number of man-hours to create, frequently are delivered with numerous problems, and are difficult to modify and/or support. The difficulty in creating and supporting large
10 applications is primarily due to the inability of existing software development paradigms to provide a simplification of the software development process. In accordance with current coding paradigms, the complexity of the software applications has been shown to increase as an exponential function of the number of different operations that are expected to be performed. Large
15 programming efforts suffer in terms of reasonable performance, reliability, cost of development, and reasonable development cycles.

[13] Object Oriented Programming (OOP) attempts to improve productivity whenever a problem can be simplified by decomposing it into a set of black-box objects. Object oriented programming depends heavily upon the benefits of
20 data hiding, inheritance, and polymorphism to simplify software design. If a network management and service provisioning solution cannot be subdivided into objects, object oriented programming does not offer significant productivity improvements. Moreover, heavy reliance on object oriented programming to achieve compact code intending to reduce the size of software
25 applications and perhaps development time, suffers from deeply nested function calls which creates a processing overhead leading to inefficient code. Deep nesting of function calls obscures the implementation paradigms used; thereby negatively impacting code debugging, code maintenance, and further development thereof.

M&C file No.

Client/Assoc. file No.

[14] Of special interest to the description herein is human-machine interface programming for information display analysts interacting with NMS systems 130. Every data network entity has operational parameters associated therewith. Associations between data network entities are also made to enable:
5 service provisioning (signalling, data transport, billing, etc.), providing redundancy (equipment, transport, bandwidth, etc.) as well as providing network management related data transport (network status updates, alarms, etc.) The ammount of configuration and/or surveillance information associated with even the simplest of managed data transport networks tends to be very
10 large. The information is typically divided up and presented to the analyst selectively in accordance with a context of network management and service provisioning. The selected information is presented to the analyst for visual inspection typically using view panels.

[15] Object oriented programming techniques are used to implement very
15 basic displayable elements known as widgets. Widgets include: icons, pull-down menus, buttons, selection boxes, progress indicators, on-off checkmarks, scroll bars, window frames, window manipulators, toggle buttons, sliders, tabs, and many other displayable elements for displaying information and for inviting, accepting, and responding to user actions.

20 [16] Prior art efforts including: Preside™ by Nortel Networks Corp., IP Manager™ by Cisco Systems Inc., OneVision Management System™ by Lucent Technologies Inc., NetProvision Activator by Syndesis Limited, Resolve 2.1 by Orchestream Holdings Plc., and others, capture the: properties, associations, relationships, functionality and management of data network entities, as well as
25 methods of interaction therewith in providing network management and service provisioning solutions into large monolithic software applications.

[17] These efforts are all laudable, but at the same time, these solutions include hard-coded user interfaces making it difficult to develop, extend and/or modify as necessary to add suport for newly developed data network

M&C file No.

Client/Assoc. file No.

entities, new data transport protocols, new network management and service provisioning technologies, etc. Further these solutions include hard-coded rules pertaining to valid actions that can be performed on data network entities as well as hard-coded rules pertaining to valid value ranges for each operational parameter thereof. The rules are part of what is known in the art as domain logic.

[18] The productivity of the development and maintenance of such complex software applications for network management and service provisioning suffers. In particular, view panels providing selective display of information have to be coded anew, the whole software application needs to be re-compiled and re-deployed to provide support for additional and/or updated data network entities. There is always a risk of causing errors in existing code when dealing with such large software applications thereby requiring extensive regression testing to verify the integrity of the existing code.

[19] Other prior art network management and service provisioning solutions include the use of element management software. Element management software is typically provided by equipment vendors to enable configuration of specific data network equipment sold. Element management software typically provides command line access to configuration of operational parameters. Such techniques suffer from the use of a large number of element management software applications, an inconsistent interface and a lack of a data network wide view of the managed data network entities.

[20] There therefore is a need to devise improved methods of software application code development and maintenance taking into account the above mentioned complexities.

M&C file No.

Client/Assoc. file No.

Summary of the invention

[21] In accordance with an aspect of the invention, graphical user interface framework for network management and service provisioning is presented. The framework implements: a file parser, methods of accessing a managed
5 object server to interact with instances of managed data network entity objects, a plurality of view panel components and a command interpreter. A plurality of run-time loadable .form files are provided to specify the functionality, look-and-feel of view panel instances used by an analyst manipulate information in interacting with the network management and service provisioning
10 environment. The view panel components are generically coded to provide support for the dynamically configured human-machine interface. The parser is used to load the .form files and instantiate view panels on demand to created listing, filtering, creation as well as configuration forms.

[22] In accordance with another aspect of the invention, a method of
15 dynamically configuring a human-machine interface is provided. The method comprises a sequence of steps. Particulars of at least one command received by a framework are extracted using a command interpreter. The particulars of the received command are made available to modules registered with the framework. At least one module determines an interaction context requiring a
20 view panel to be displayed. The view panel may include a combination of view panel components. A descriptor file associated with the interaction context is selected to configure instances of view panel components. And the view panel is displayed.

[23] Advantages are derived from a separation between network
25 management and service provisioning software application coded functionality and the specification of the interface presented to the analyst. The separation also provides for an efficient development, maintenance and deployment of the network management and service provisioning solution. The modularity provided results in a relaxed dependancies among source files, among

M&C file No.

Client/Assoc. file No.

development efforts to enable a faster parallel development, particularly applicable to symplify the development of network management and service provisioning solutions while reducing requirements for regression testing.

Brief description of the drawings

- 5 [24] The features and advantages of the invention will become more apparent from the following detailed description of the preferred embodiment(s) with reference to the attached diagrams wherein:

FIG. 1 is a schematic diagram showing data network elements implementing connected data transport networks;

- 10 FIG. 2 is a schematic diagram showing elements implementing a network management and service provisioning solution in accordance with a preferred embodiment of the invention;

- 15 FIG. 3 is a schematic diagram showing a managed entity object hierarchy used in providing the network management and service provisioning solution in accordance with the preferred embodiment of the invention;

FIG. 4 is a schematic diagram showing an managed entity containment hierarchy used in providing the network management and service provisioning solution in accordance with the preferred embodiment of the invention;

- 20 FIG. 5 is a schematic diagram showing an exemplary human-machine interface associated with a software application, in accordance with an embodiment of the invention, participating in a network management and service provisioning solution;

FIG. 6 is a schematic diagram showing, in accordance with an exemplary embodiment of the invention, associations of data entities implementing a

M&C file No.

Client/Assoc. file No.

software component enabling a dynamically configurable human-machine interface;

FIG. 7 is a schematic diagram showing, in accordance with an exemplary implementation of the invention, a dynamically configurable view panel in support of a dynamically configurable human-machine interface;

FIG. 8 is another schematic diagram showing, in accordance with an exemplary embodiment of the invention, associations of data entities implementing another software component enabling a dynamically configurable human-machine interface; and

FIG. 9 is another schematic diagram showing, in accordance with an exemplary implementation of the invention, another dynamically configurable view panel in support of a dynamically configurable human-machine interface.

[25] It will be noted that in the attached diagrams like features bear similar labels.

15 **Detailed description of the embodiments**

[26] FIG. 1 is a schematic diagram showing data network elements implementing connected data transport networks.

[27] Data network nodes 102, 110, 120 are physically interconnected in the data transport network 100 via physical links 108. Data transport networks 100 may be bridged via bridge data network nodes 104 to enable data exchange therebetween. Connected data transport networks 100 can be grouped defining areas of focus and influence for the purposes of network management and service provisioning, known as network partitions 106.

[28] Physical links 108 provide Open Systems Interconnection (OSI) Layer-1 connectivity between data network nodes 102/104/110/120 physically

M&C file No.

Client/ Assoc. file No.

conveying data for OSI Layer-2 data links between nodes 102/110/120 end-to-end. A Layer-2 data link may be provisioned over at least one physical data link 108 – the sequence of physical data links 108 used representing an OSI Layer-3 path 128.

- 5 [29] Network management and service provisioning is typically performed with the aid of at least one Network Management System (NMS) 130 connected to at least one node 102 associated with a data transport network 100.

[30] FIG. 2 is a schematic diagram showing elements implementing a network management and service provisioning solution.

- 10 [31] In accordance with a preferred embodiment of the invention, a framework 220 is provided. The framework 220 may include a combination of hardware and software application code. The framework 220 facilitates the implementation of a software development methodology for coding complex software applications 210 relating to network management and service
15 provisioning.

[32] The framework 220 implements a new architecture for providing network management and service provisioning solutions. The new architecture categorizes the above presented elements into:

- Manageable data network entities representative of field installed
20 managed data network entities to be configured and controlled in providing network management and service provisioning solutions. The managed entities include:

i. Physical data network equipment installed in the field such as: nodes 102/104, routers, switches, hubs, OC-3 links 108, etc., and

- 25 ii. Logical data network entities associated with data network equipment installed in the field such as: network partitions 106, paths 128, virtual circuits, virtual routers etc.;

M&C file No.

Client/Assoc. file No.

– Network management and service provisioning software applications 210 used to configure and control the manageable data network entities. The software applications 210 include as mentioned above: inventory reporting 214, configuration management 212, statistics gathering, performance reporting, fault management, network surveillance, service provisioning 216, billing & accounting, security enforcement, etc.;

– Network management enabling technologies 230 providing interaction with manageable entities such as logical and field installed physical managed data network entities. Enabling technologies 230 include:

i. Data network management and service provisioning protocols: SNMP, CMIP, CLI, DNS, etc., and

ii. Data network management and service provisioning devices: databases, DNS servers, etc.

The interaction may be command driven as specified by the software application 210, as well as event driven as a current state of the managed data transport network(s) in the realm of management changes.

[33] The enabling technologies 230 include support for a concept known as “persistence”. Each data network entity including data network equipment has an associated group of parameters. These parameters either have an effect on the operation of the data network entity or label the data network entity. The persistence concept encompasses the storage of, access to, reading, writing, modifying, synchronization/reconciliation, etc. of persistence parameters to control the operation of data network entities.

[34] The persistence parameters can be stored in a network management and service provisioning database 132, as well as in registers associated with the managed physical data network equipment installed in the field. The persistence access to, reading of, writing of, modification of these parameters is

M&C file No.

Client/Assoc. file No.

provided via the data network management and service provisioning protocols mentioned above. Persistence reconciliation and synchronization is performed between a persistence database and a persistence value held in a volatile register ensuring a correct record keeping thereof, fast access to the persisted
5 information and backup thereof. Distributed storage of persistence information is also used in reconfiguring data network equipment subsequent to network failures.

[35] The persistence concept also encompasses special persistence types such as: constant persistence which can only be initialized but not modified or
10 written to subsequently; as well as derived persistence which is not stored but rather calculated from other persistence values when needed.

[36] Further information regarding persistence entity support is provided in co-pending U.S. Patent Application bearing Attorney Docket
Number 11922-US filed on even date entitled "NETWORK MANAGEMENT
15 SYSTEM ARCHITECTURE" which is incorporated herein by reference.

[37] In accordance with the preferred embodiment of the invention, the framework 220 does not interact with the database 132 directly but rather through a Managed Object Server 200. The MOS 200 makes use of a managed data network entity object derivation hierarchy 300 to instantiate 202 and
20 interact 204 with managed entity object instances 206. An exemplary managed object derivation hierarchy is presented in FIG. 3. Instantiated managed entity objects 206 form a containment hierarchy 400 an example of which is presented in FIG. 4.

[38] In accordance with a preferred embodiment of the invention, a group of
25 generic view panel components 240 are defined to provide support for a dynamically configurable human-machine interface. The generic view panels components 240 provide support for the display of selected context-sensitive information for inspection by the analyst as well as provides front end

M&C file No.

Client/Assoc. file No.

processing of input. The generic view panel components include, but are not limited to:

- an entity listing component enabling the presentation of data network entities having common characteristics;
- 5 – an entity list filtering component enabling the sifting of listed entities by an entity listing component to hide unwanted list elements;
- an entity tree creation component presenting (hierarchical) associations and relationships between data network entities for navigation through connectivity information;
- 10 – an entity parameter inspection component providing operational parameter browsing and editing – entity parameter inspection component derived view panels may also be used for data network entity context-less configuration;
- 15 – an entity inclusion/creation component for instantiation of managed data network entity objects managed via the MOS 200;
- an entity configuration component for changing a state of at least one data network entity in the realm of management of the network management and service provisioning solution – entity configuration component derived view panels are used for data network entity configuration subject to a context;
- 20 – a validation component for interacting with the MOS 200 in validating ranges of parameters selected;
- a commit component for interacting with the MOS 200 in implementing configuration changes; etc.
- 25 [39] In accordance with the invention, the generic view panel components are combined, instantiated and configured at run-time to provide the analyst with a context specific interface for interaction therewith. Other generic view panel components may be developed independent of or in combination with the above mentioned components to extend the functionality provided by the
- 30 framework 220. For example a statistical information display component may be used in presenting statistical information to the analyst.

M&C file No.

Client/Assoc. file No.

[40] FIG. 6 is an exemplary generic abstract implementation of a view panel component enabling entity listing in support of a dynamically configurable human-machine interface. In particular, the presented implementation of the entity listing component 600 is a combination of view panel components inheriting functionality of the entity list filtering view panel component. Concrete derived entity listing view panel components 610, 620 and 630 are also shown. FIG. 7 is a schematic diagram showing, in accordance with an exemplary implementation of the invention, a generic dynamically configurable view panel used in listing data network entities subject to filtering options in support of a dynamically configurable human-machine interface. The generic view panel shown corresponds to an instance of the entity listing view panel component 630.

[41] FIG. 8 is another exemplary generic abstract implementation of a view panel component enabling entity configuration in support of a dynamically configurable human-machine interface. FIG. 9 is another schematic diagram showing, in accordance with an exemplary implementation of the invention, another generic dynamically configurable view panel used in configuring data network entities in support of a dynamically configurable human-machine interface.

[42] In accordance with the preferred embodiment of the invention, a group of run-time loadable view panel configuration description ".form" files 226 are provided. Each one of the .form files 226 is a human-readable source code file consolidating all relevant particulars a particular view panel used with respect to a particular network management and service provisioning solution. Each .form file 226 specifies: the look-and-feel of an associated view panel, widget attributes, etc.

[43] In accordance with the preferred embodiment of the invention, the .form files 226 may be edited at run-time and re-loaded to change the functionality, presentation, etc. of the associated view panels. For this purpose the

M&C file No.

Client/Assoc. file No.

framework 220 includes a .form file parser 222. The use of the parser 222 at development, provides a troubleshooting tool for interface related issues in coding network management and service provisioning solutions without the need to re-compile and re-link edited code. The use of the parser 222 provides
5 for modifying and extending deployed network management and software provisioning solutions at reduced costs.

[44] Other benefits provided by the parser 222 and the .form files 226 are realized from an generalized improved ability to internationalize the interface presented to the analyst. The internationalization of the interface refers to the
10 presentation of the NMS 130 interface in a written language specific to the analyst interacting with the network management and service provisioning solution.

[45] In accordance with an exemplary implementation of the invention, the .form files 226 are written language specific and include labeling strings for
15 each written language supported. Additional written languages may be supported via additional .form files 226. Although useful, this implementation require a coding engineer that is also versed in the specific language for which the language specific .form file 226 is coded.

[46] However, in accordance with the preferred embodiment of the
20 invention, the .form files 226 are written language independent providing run-time replaceable displayable string placeholders. Subject to a global register holding a specification of a specific written language to be used, each string placeholder is populated at run-time with a currently specified written language displayable strings. In accordance with the preferred embodiment of
25 the invention, displayable strings associated with a written language are provided via a run-time loadable 228 ".bla" files 227.

[47] Examples of network management and service provisioning context specific view panel specifications provided via .form files 226 include but are

M&C file No.

Client/Assoc. file No.

not limited to: create/configure/list/filter IP route, create/configure/list/filter IP link, create/configure/list/filter Label Switched Path (LSP), configure/list/filter IP router, configure/list/filter Label Switching Router (LSR), etc.

5 [48] In accordance with the preferred embodiment of the invention, the parameter inspection view panel component used in combination with the list and filter view panel components may be used to configure newly developed data network entities associated with the MOS 200 in a context-less fashion while pending development of context specific view panels. The development
10 of context specific view panels includes the writing of at least one new .form file 226 associated with the newly developed data network entities.

[49] In accordance with the an exemplary embodiment of the invention, the functionality provided by view panel components 240 presented above is further combined to define Graphical User Interface (GUI) modules 260. An
15 exemplary GUI module 260 is an Object Navigate Module 500. An exemplary dynamically generated view panel presented to an analyst interacting with the Object Navigate Module 500 is presented in FIG. 5. The Object Navigate Module 500 combines functionality of the tree 510, list 520, filter 530 and attribute 540 components.

20 [50] In accordance with the invention, the GUI modules 260 are coded in run-time loadable fashion enabling on demand loading and registration thereof with the framework 220. Additional GUI modules 260 may be developed, the modularity provided requires less time to develop and relatively less susceptible to break other code when compared to prior art solutions.
25 Therefore less regression testing is required by implementing the methods presented herein.

[51] In operation, the analyst interacts with the NMS system 130 and in particular the analyst interacts with displayed widgets. Input from the analyst

M&C file No.

Client/Assoc. file No.

is packaged in the form of commands and the commands are forwarded to the framework 220. The framework 220 inspects each received command using a command interpreter 224. Upon extracting particulars specified in a command, the framework 220 makes the particulars available to the registered
5 modules 260.

[52] Each registered module 260 having access to the particulars of the received command, determines whether to process the received command. On determining to process the received command, each particular module 260 makes a further determination whether the processing of the received
10 command necessitates the instantiation of view panel components. The instantiation of view panel components uses the .form files 226 to configure the view panel components and widget attributes, to display a combination of widgets having a behavior specified therein to enable the processing of the received command.

15 [53] In accordance with the invention, domain logic 208 is not coded in the .form files 226 but is made available by the MOS 200. In validating and committing changes to operational parameters, commands are issued by the module 260 to the MOS 200. The MOS 200 having access to viable operational parameter ranges as well as operational parameter visibility implements
20 changes to operational parameters subject to the domain logic 208.

[54] Operational parameter visibility rules 208 are of an important mention herein for the analyst interaction is subject to a scope-of-command and a span-of-control implemented using these rules 208. On interacting with the network management and service provisioning environment, the analyst is
25 authenticated and subsequently authorized to perform specific actions defining the scope-of-command – the actions being allowed on specific data network entities at specific times under specific conditions defining the span-of-control.

M&C file No.

Client/Assoc. file No.

[55] Distributed computing techniques are used in exchanging messages carrying commands between the framework 220 and the MOS 200. A Common Object Request Broker Architecture (CORBA) bus 250 is preferably used to provide support for a distributed computing environment.

5 [56] The MOS 200 brokers access to connectivity information, and perhaps to statistical information, held in the database 132. The separation between the GUI modules 260 and the database 132 enables an independent development of enabling technologies 230 (schema) while enabling a generic human-machine interface development. Further the use of the framework 220 and the MOS 200
10 provides developers with a freedom to develop GUI modules 260 and associated software applications 210 without hardcoded knowledge of manageable data network entities.

[57] In accordance with the invention, common functionality such as managed entity selection, selection lists creation and manipulation, visual
15 labeling of selected managed entities is provided as services associated with the framework 220. The functionality of these services provided by the framework 220 transcends the functionality of modules 260 and software applications 210. As an example, these services for example enable the selection of data network entities via the Object Navigation Module 500, visual labeling thereof for
20 display on a human-machine interface associated with the service provisioning application 216.

[58] The .form files 226 capture functionality of the specified view panels as well as a methodology used in separating the information displayed to the analyst in a concise, efficient, context sensitive manner. An exemplary
25 methodology in detailed in Appendix 1.

[59] The embodiments presented are exemplary only and persons skilled in the art would appreciate that variations to the above described embodiments

M&C file No.

Client/Assoc. file No.

may be made without departing from the spirit of the invention. The scope of the invention is solely defined by the appended claims.

M&C file No.

Client/Assoc. file No.

Appendix 1

[60] There are numerous configuration forms within the NMS 130 developed by many different developers, in many different groups. In the future even more diverse numbers of groups could develop features for the NMS 130 and be responsible for new configuration forms. It is essential for new development be undertaken in a consistent manner. Consistency presents a professional looking product to the analyst as well as reduce training times, allow operations within the NMS 130 to be undertaken in less time and limit the amount of customer support which vendors will be required to provide.

[61] It is important that the order and layout of widgets within the window match the workflow of the analyst. Text fields and labels, widget selection, filtering tasks, etc., are to be positioned within the screen so that the analysts do not have to undertake excessive mouse movements or visual searches to accomplish tasks. The positioning and labeling of repeated items or item types within the different configuration forms are required to remain consistent between forms. Consistency between forms builds a confidence that will help analysts navigate through new configuration forms with fewer errors, at greater speeds and with less training.

[62] Configuration Forms are intended not to have menus and tool bars. All required commands for configuration forms should be accessed through the use of command buttons, either in a command button column or positioned next to specific areas within the form that is active. Command buttons provide the analyst with unhidden, accessible options to undertake actions within the configuration form.

[63] Generic configuration windows should consists of the following standard items or areas:

[64] The title bar contains the name of the configuration form and generally specifies the managed entity that is being configured.

M&C file No.

Client/Assoc. file No.

- [65] This area contains information fields that, when filled in, will configure different parts of the network or different aspects of the network management and service provisioning solution. This section could contain standalone command buttons for undertaking specific tasks related to widgets within the view panel.
- [66] Generally, the configuration fields area should be presented as a tabbed pane that will allow view panel developers the flexibility of breaking up the fields that are required for a configuration into manageable amounts for the analyst. Also, it provides a possible simple way to add information to the form in the future.
- [67] Nested tab within the configuration forms are not preferred. However, tabbed features within forms can be implemented without requiring the entire form to be tabbed.
- [68] The command button column should preferably have a vertical orientation and should be placed at the top right of the form.
- [69] Command buttons can be located outside of the Command Button Column. This allows buttons to be associated with the specific areas of the view panel that they influence.
- [70] A Message Bar, or text display, may be utilized for help or action notes specific to the configuration form which is being filled in. An example would be: if a value being entered into a field is out of range, it could be highlighted in this area.
- [71] A Progress Bar is preferably presented within the view panel to indicate that work is being done. Typically the progress bar is to be used for tasks that take 10 seconds or more, such as downloading information or filtering values.
- [72] Often features may be disabled (grayed out) or completely hidden to prevent analysts from accessing them (scope-of-command/span-of-control).

M&C file No.

Client/Assoc. file No.

This happens for various reasons such as: not having appropriate access permissions to operate certain features, not having the input correct or enough information to undertake an action, not having checked off the preference which activates a feature etc.

[73] In general, if the analyst could under different circumstances use a feature, then it is appropriate to gray it out. The analyst will be aware of the feature's existence and understand that actions are required to get access thereto. Grayed-out features are also valuable because they allow the analyst who is exploring an unknown program to see that they exist and have functionality.

[74] If the analyst does not have access to features and would never have access or reason to use them, then it is appropriate not to show the feature at all. Grayed-out features which will never be used clutter the view panel, confuses the analyst and possibly increases the time taken to access the features which can be used (i.e., the analyst is required to scroll / tab through grayed-out menus, tabs and buttons etc.)

[75] The configuration view panels will typically use two forms of error warnings:

- When information is entered into or selected for a field and it is of the wrong type, the configuration form will not accept the entry and a message will typically be displayed at the bottom of the screen in a message bar.
- When incorrect values have been entered and the

[76] "Ok", "Apply" or any other confirmation button is clicked, an alert error dialog window may pop-up advising the analyst that problems have been encountered.

[77] Alert boxes convey messages and warnings to the analyst. They contain window titles, symbols, message headers, messages and command buttons.

M&C file No.

Client/Assoc. file No.

[78] Error alert boxes display a message describing a major problem that has been encountered and requires the analyst's intervention or correction in order to continue. Error alert boxes contain an icon representing the severity of the problem, a textual description of the error, the error number in the title and typically one button. Error alert boxes may also contain controls to correct the error. If the error cannot be corrected from the dialog box, a Close button is preferably presented. A Help button is optional (when deemed appropriate).

[79] A button representing the corrective action may be presented if the analyst can correct the error, followed by an escape button such as Cancel, and finally Help (if deemed appropriate). Cancel would close the error alert box and cancel the operation that initiated the error.

[80] Error alert boxes like all alert boxes are created automatically by the framework 220.

[81] Other alert boxes can result from the use of configuration forms. Following is a brief list of the different types:

[82] Informative alert boxes are used only for conveying information and only contain a Close button. These display a body of text typically.

[83] Warning alert boxes prompt the user to confirm an operation. These contain a textual description and minimally two buttons. A first button typically is a Continue or Yes button that reflects the action about to be performed. A second button allows the user to back out of an operation and would typically be labeled Cancel or No, depending on the message context. If appropriate, a Help button can be added as a third button.

[84] A default button should not be set if the action about to be performed has potentially destructive consequences.

[85] Generic configuration forms may use two kinds of progress indications:

M&C file No.

Client/Assoc. file No.

- Progress Indicator Bar is typically located at the bottom of the configuration form. The progress bar typically appears while filter or look-up tasks are being performed.
- The Progress Dialog Box, would only appear after the configuration forms inputs have been applied through the use of an "OK" button (or equivalent). It is only to be used on those configuration forms that require a long period of time to implement previously entered information.

[86] A Progress Indicator is typically located on the right side of the status bar. A progress indicator is used in the status bar if the application performs frequent, non-interruptive operations, likely to last more than 10 seconds. A non-interruptive operation would be one that does not require the analyst to stop working until it is complete.

[87] If it is impossible to determine the progress of the activity towards completion (i.e., the completion point is unknown, as in a database query), then the use of a looping progress bar is preferred. A text pane can be used alongside the progress bar to give feedback on what the progress bar is indicating.

[88] If the rate of progress towards a known completion value is known, then the use of a non-looping Progress Bar is preferred. It is preferred to indicate the percentage complete within the progress bar itself. A text pane next to the Progress bar may be used to give additional information and feedback.

[89] A progress dialog may be used to modally display the progress of a process that the application is undertaking. It is to be used if the analyst cannot continue working until the process is done (i.e., they cannot multitask within the application). For example using a progress bar within the status bar of the primary view panel would be appropriate if the application has sent a query to a remote database, allowing the client machine to continue working. However, an action such opening a large file on the client machine would not allow for the

M&C file No.

Client/Assoc. file No.

analyst to perform other tasks, so a progress dialog would be appropriate. Progress dialogs are to be used for expected delays of more than 10 seconds. An hourglass pointer is preferably used if the delay is less than 10 seconds.

[90] If there are multiple milestones on a long process, text updates are preferably used to keep the analyst informed.

[91] A progress bar should contain a method of halting the task at hand. For this reason, all progress dialogs have either a Cancel or Stop button. Cancel is used if halting the process will result in returning the application to the same state as before the action was taken. Use Stop if the process changes the state of the application as it is performed (for example, deleting files, generating a list, etc.), or if data loss may occur. In the former case, stopping will halt any remaining changes, but will keep those performed up to that point. In the latter case, stopping should be followed with a Warning Alert box informing the analyst that there may be a loss of data.

[92] Progress Dialogs automatically close when the action being performed is completed.

[93] Help buttons are preferably supplied within the different configuration forms where applicable.

[94] Tool tips are used to describe, or provide greater detail, on a widget under the system's pointer immediate location. Tool tips can be associated with widgets such as: buttons, sliders, combo boxes, icons, etc. If there is a keyboard shortcut for activation of the particular widget then the keyboard shortcut should be displayed in the tool tip.

[95] Tool tips should be used carefully, and only if they add information. The tool tip expands upon the widget's purpose. If a text label associated with the widget is specific enough, the use of a tool tip is not recommended. Tool tips presenting long, complex information (paragraphs of text, lists, a matrix of information, etc.) should be avoided.

M&C file No.

Client/Assoc. file No.

- [96] Tool tips should appear after a system pointer hovers over a widget for more than approximately 750 milliseconds and its display should not last for more than 4 seconds.
- [97] Slightly longer of contextual help can be given in a message bar at the bottom left of the view panel as an alternative to the use of tool tips. The analyst is typically given the option to turn this feature off in a preference menu.
- [98] All actions that are required within a configuration form can be mapped to command buttons. There are two different types of command buttons, those located in the command button column at the top, right side of the form and those located next to objects within the form. Command buttons are a fast, direct and simple way for the user to interact with the view panel.
- [99] Command buttons are buttons within a configuration form that allow the analyst to undertake operations. Buttons that have general influence are preferably placed vertically in the command button column at the top right side of the view panel. Buttons that influence specific objects within the view panel should be located proximally to that object.
- [100] A OK button is used when a dialog box requires the user to input information, make changes, etc., but the information is not submitted on the fly, i.e., the changes do not take affect until the analyst decides to save or employ them by pressing OK. If OK is used, no mnemonic is applied. Actuation of the OK button will result in the changed or entered data within the form being saved and the form being closed.
- [101] A dialog window can be used to ask the analyst if the "OK" action should be undertaken as well as to indicate if changes were made to the object being configured simultaneously by another analyst.
- [102] In configuration forms the Save button is used to submit changed or new data entered into a form without closing the form.

M&C file No.

Client/Assoc. file No.

[103] The Close button is used in a dialog box that does not require the analyst to input any information, or the input is performed on the fly. Examples would be in an information alert box that just displays a text message to the user, or a dialog that contains a slider bar whose values are translated to the main application on the fly.

[104] Cancel is used in the same dialog boxes where an OK button occurs. It closes the window without applying the user's changes or input. Cancel does not have a mnemonic applied to it.

[105] The Help command button launches context sensitive online documentation.

[106] A command button can be set as the default button for a configuration form. That is, pressing return will trigger the button. Default buttons add speed and convenience to dialog boxes. However, the use of a default button is not recommended if the action is a destructive one. Default buttons do not have mnemonics. Default buttons have a thicker border surrounding them.

[107] Command buttons which when actuated open view panels, should have ellipses, placed after their label text.

[108] Command buttons with ellipses are appropriate for opening view panels that the analyst can choose something from, such as selecting path endpoints. Also, they can be used for bringing up centralized functions, such as connection scheduling or filtering resources for network information. However, command buttons with ellipses are not intended to be replacements for tabs within configuration forms.

[109] Stand alone command buttons are located next to the objects in the view panel which they directly influence. If a command button only influences one item within the view panel then it should be located next to that item.

[110] Listed below are a selection of widgets.

M&C file No.

Client/Assoc. file No.

- [111] An editable field is used to input alphanumeric data into a configuration form. Once the text field is clicked or tabbed to, a blinking insertion point indicator appears within the field.
- [112] The editable field is also used for strings of editable text.
- [113] A non-editable text field is used to show alphabetical, numeric, and alpha-numeric data to the analyst. The analyst can copy the information within the field, but the field cannot be modified.
- [114] Non-editable text fields are used for strings of read only textual information that are generally less than 40 characters long.
- [115] A text area is used to input large amounts of alphanumeric data into a configuration form. A text box size used should be wider than it is tall (a rectangle) and of a useful width (30 characters) so that it can be read without spreading the information over too many lines. The text area is typically used when many sentences or more than one line of text are required.
- [116] A slider scale is a visual way for the analyst to select a value within a configuration form which usually is qualitative in nature. The slider should be made up of a label, followed by a slider, and finally followed by an editable field (which is optional depending on the nature of the values being entered). The slider scale is typically used to select a value from a range of values or states.
- [117] A table is a titled grid of information used to show large sets of data which are required to be entered into a configuration form.
- [118] A check box list is composed of an over all label positioned next to a series of check boxes that are each associated with a label.
- [119] The check box is a preferred way of selecting multiple items from a list. It provides the analyst with a flat view of all the options facilitating utility and expediency within the view panel.

M&C file No.

Client/Assoc. file No.

[120] A radio button is the preferred way of selecting a single item from a list. It provides the analyst with a flat view of all the options facilitating utility and expedience within the view panel.

[121] A scroll pane enables the analyst to control the view of what is being displayed at any time. It consists of a vertical or horizontal "channel" through which a scroll box is moved. Each end of the channel has a scroll arrow.

[122] Configuration form scroll bars are used only within widgets (e.g. combo boxes etc.)

[123] A group box is a non-interactive widget which associates related items within configuration forms (e.g. they all control the same data network entity). The group box has a label and a border.

[124] Group boxes should only be used where a distinct group of items require separation from the rest of the form. Group boxes should almost never be used to surround one item.

[125] A list box is a box containing items which can be selected or multi-selected. It can contain check boxes or radio buttons as well as scroll bars. The list box is used when the analyst is required to make multiple selections of a list of options.

[126] A combo box is an editable or non-editable value field that has a drop-down arrow and drop-down list of possible selections associated with it.

[127] Following are typical uses of the combo box:

- The Combo Box is used to select a single item from a group of items.
- In non-editable combo box's choice is limited to the elements in the drop-down list.

M&C file No.

Client/Assoc. file No.

- In editable combo boxes the analyst can choose from the drop-down list or enter a unique value in the field. Providing a list of typical selections speeds up the time it takes for the analyst to complete configuration forms.

[128] A spin box is a widget which allows the user to select a value for a field by clicking on up and/or down arrows which are associated therewith.

[129] Spin boxes may have the following drawbacks if used incorrectly:

- Spin boxes increase the chance of errors.
- The up and down arrows are small, hard to hit targets.
- The analyst is prone to getting lost if the set of data that is being selected through the spin box is too large.

[130] A Tab is a way in which the form developer can add multiple panes of information in one configuration form. Tabs preferably should run along the top of the areas that they affect. The tab that is being viewed is to be shown connected to the pane, while the ones not being viewed are in the background. The analyst is required to click on a tab to view the contents of the associated pane. All changes made on multiple tabs will be saved upon the analyst's confirmation by clicking the "OK" (or "Apply", "Save" etc.) command button.

[131] When a series of distinct groups of objects to configure exist and there is not enough room to put them all on one pane, tabs can be used to break up the groups and allow the analyst to inspect and configure them one at a time.

[132] A Paste button is used to paste the Selected Object from the Selected Object Window into an area. The Paste button is used in any situation that requires the analyst to paste information from one area to another.

M&C file No.

Client/Assoc. file No.

I/WE CLAIM:

1. A graphical user interface framework for network management and service provisioning, the framework comprising:
 - a. a plurality of generic view panel components, each view panel component being used to display information to an analyst as well as to receive input;
 - b. a parser used to load a plurality of view panel descriptor files, each descriptor file including information configuring at least one of the generic view panels;
 - c. at least one access method implementation, the access method being used by the framework to access a managed object server in interacting with instances of managed data network entity objects; and
 - d. a command interpreter used by the framework to process received command messages
 - e. a set of rules defining valid interactions with attributes of objects managed by the object management workstation; andwherein the framework is used to configure at least one generic view panel in accordance with a specification detailed in an associated descriptor file to display an instance of a configured view panel on a display associated with a network management system.
2. A framework as claimed in claim 1, wherein the framework is further operable to designate a particular generic view panel and select a particular descriptor file dependent upon received commands.

M&C file No.

Client/Assoc. file No.

3. A framework as claimed in claim 1, wherein the framework is operable to enable the registration of modules adapted to interwork with the framework.
4. A framework as claimed in claim 3, wherein the modules include JAVA plug-in modules and the framework includes a JAVA framework.
5. A framework as claimed in claim 1, wherein the information included in the descriptor files comprises a specification of widget attributes.
6. A framework as claimed in claim 1, wherein the information included in the descriptor files comprises a specification of widget attribute rules.
7. A framework as claimed in claim 6, wherein the widget attribute rules included in the descriptor files comprise indications of analyst actions allowed on widgets.
8. A framework as claimed in claim 1, wherein the descriptor files are extensible markup language (XML) files.
9. A framework as claimed in claim 1, wherein at least one generic view panel includes an input error handling routine for responding to an input error.
10. A method of providing a dynamically configurable human-machine interface, the method comprising steps of:
 - a. extracting particulars a command received by a framework using a command interpreter;
 - b. making the particulars of the received command available to registered modules with the framework;
 - c. at least one module determining an interaction context requiring a view panel to be displayed, the view panel comprising a combination of view panel components;

M&C file No.

Client/ Assoc. file No.

d. selecting a descriptor file associated with the interaction context to configure instances of view panel components; and

e. displaying the view panel

wherein the run-time configuration of view panel components enables a dynamically configurable human-machine interface.

11. A method as claimed in claim 10, wherein the method further comprises a step of accepting input.
12. A method as claimed in claim 11, wherein the method further comprises a step of validating input.
13. A method as claimed in claim 12, wherein the method further comprises a step of committing configuration changes.
14. A method as claimed in claim 10 wherein the method further comprises steps of exchanging messages encapsulating commands with a managed object server.

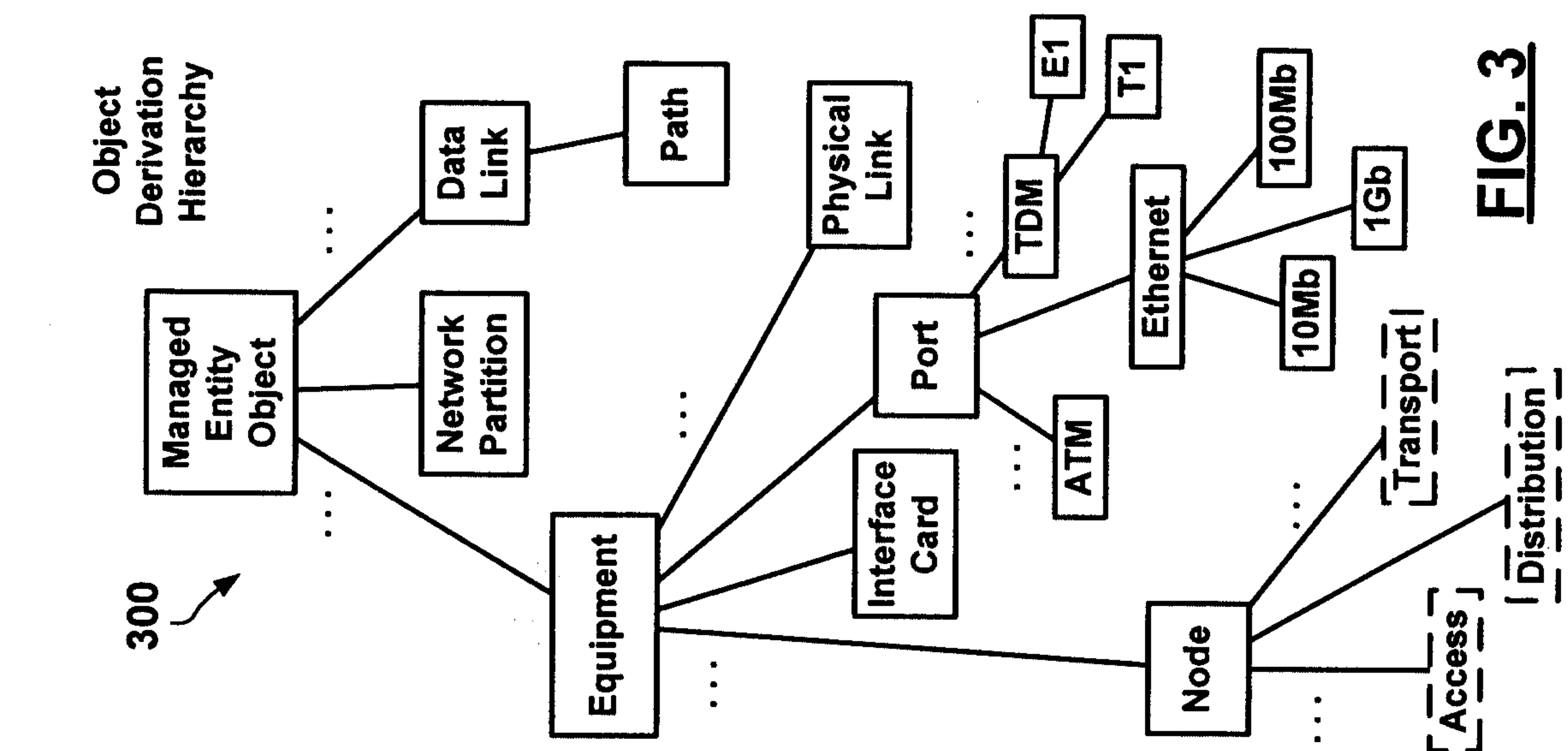


FIG. 1

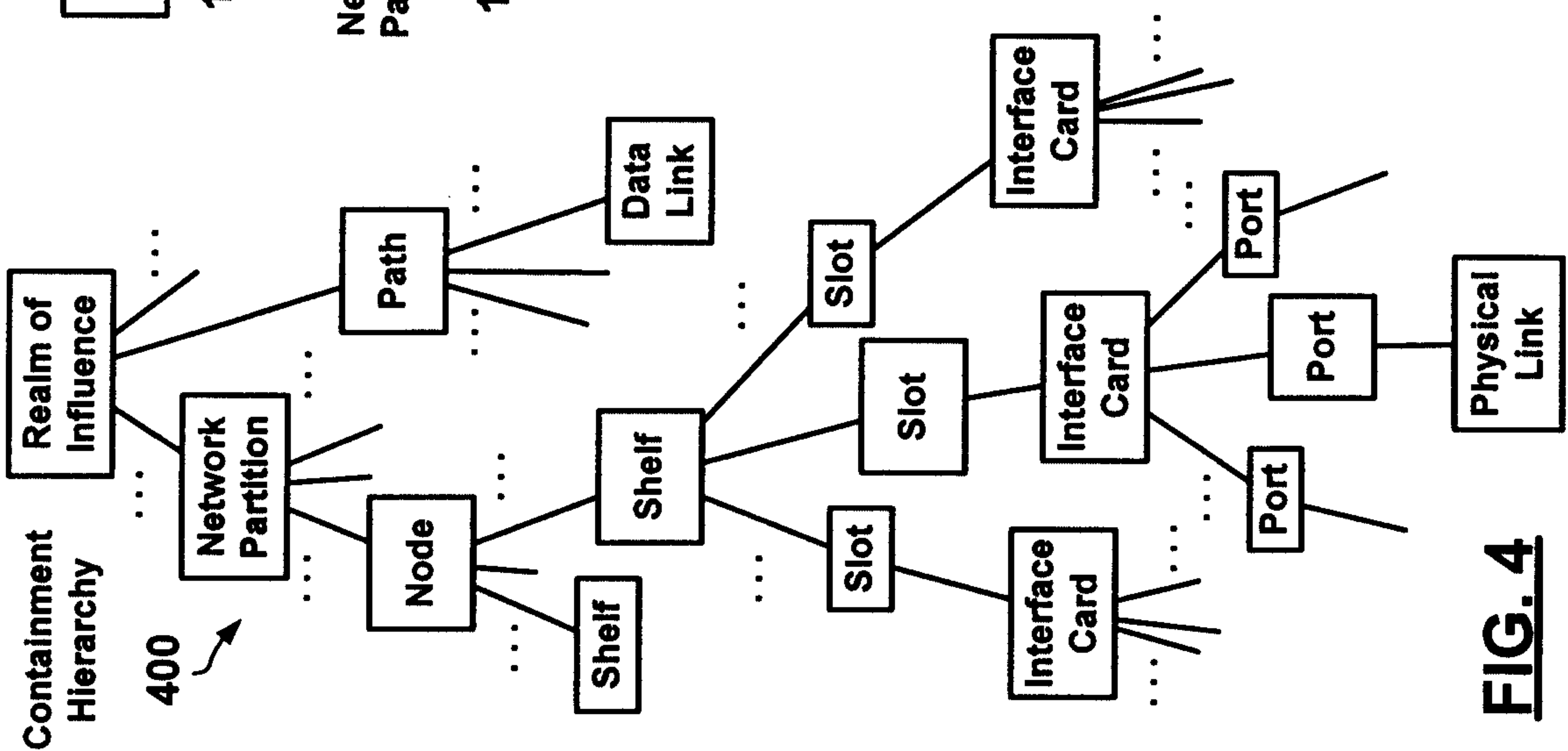


FIG. 2

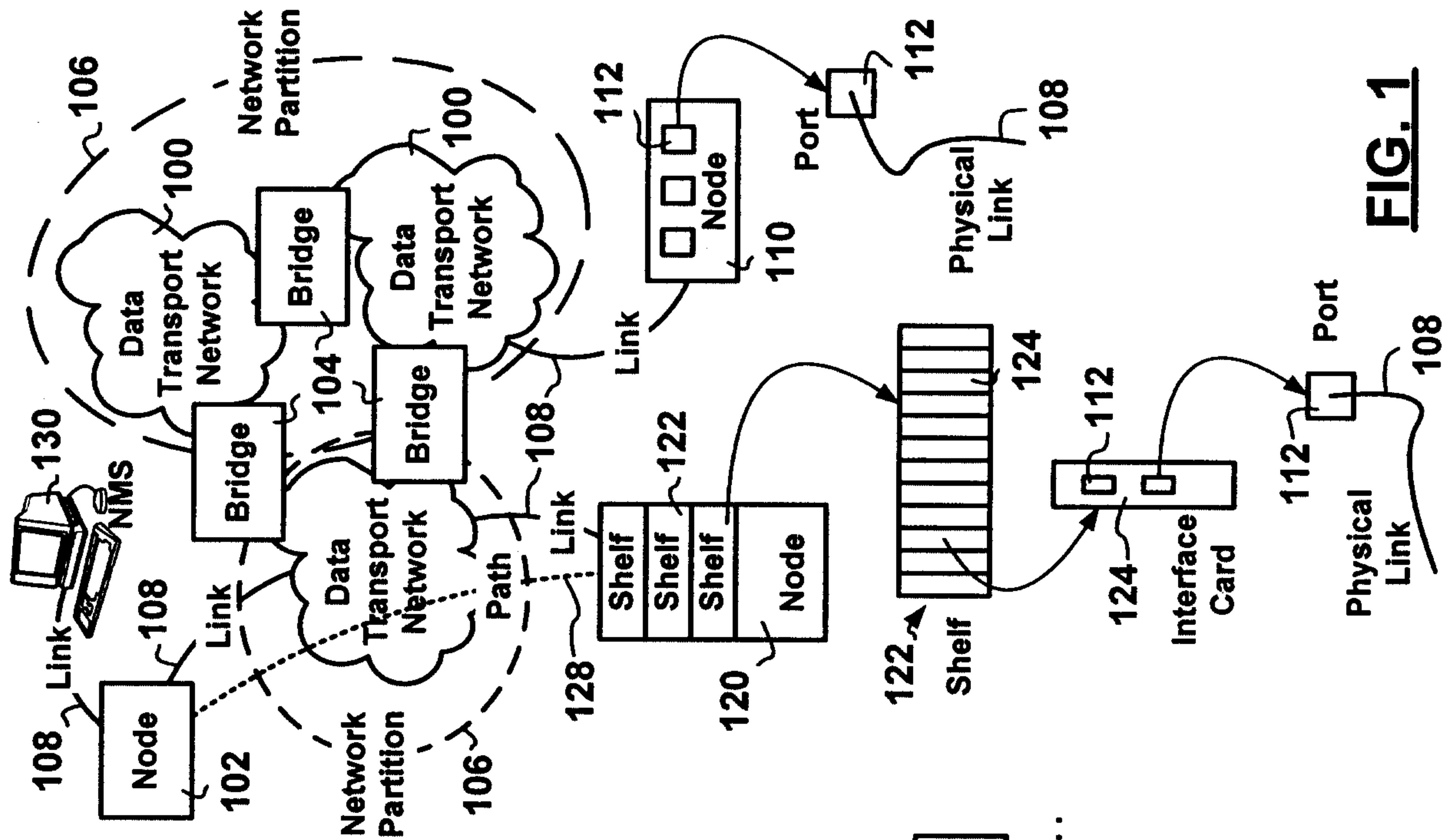


FIG. 3

Mark E. Clark

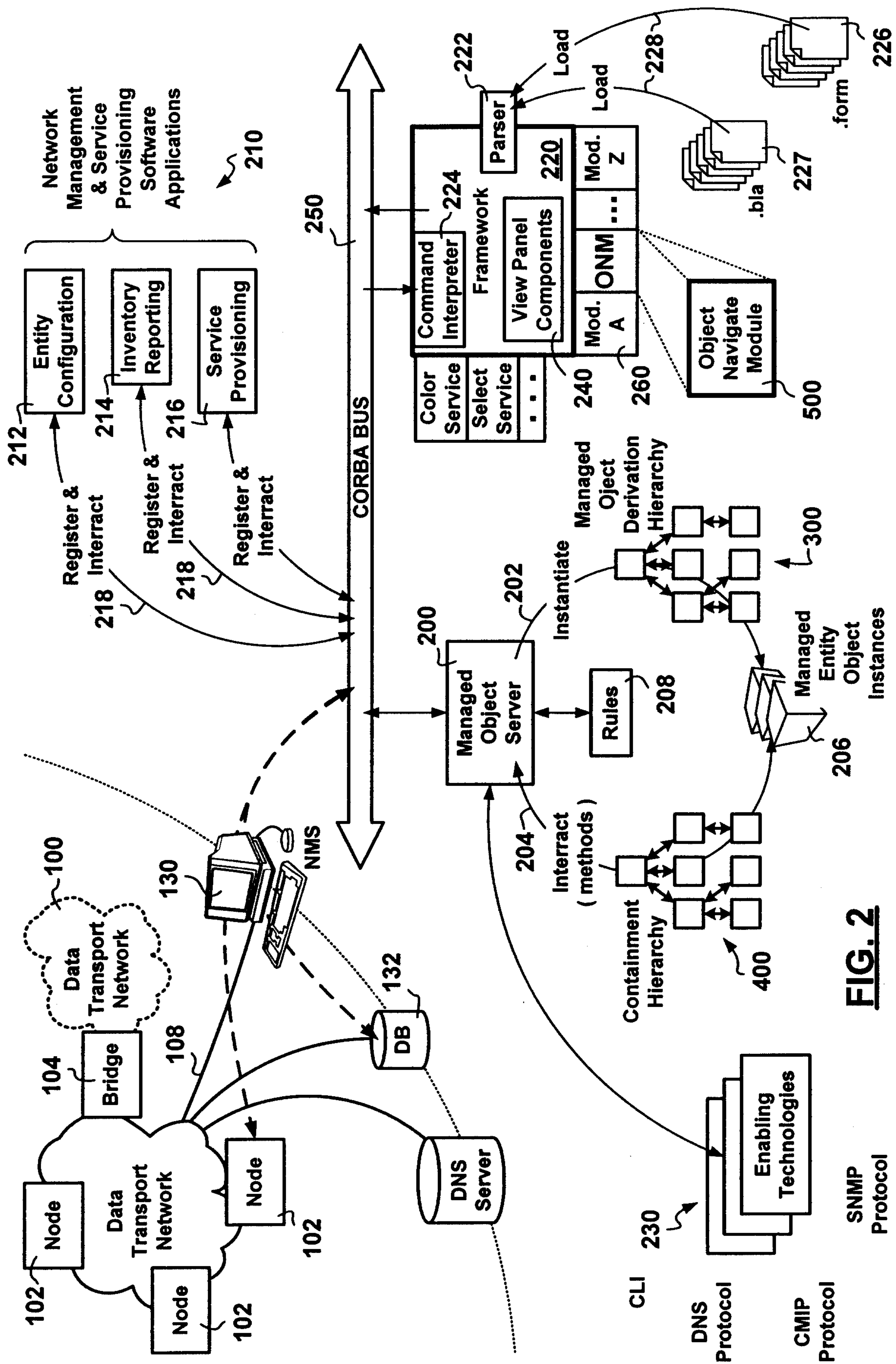
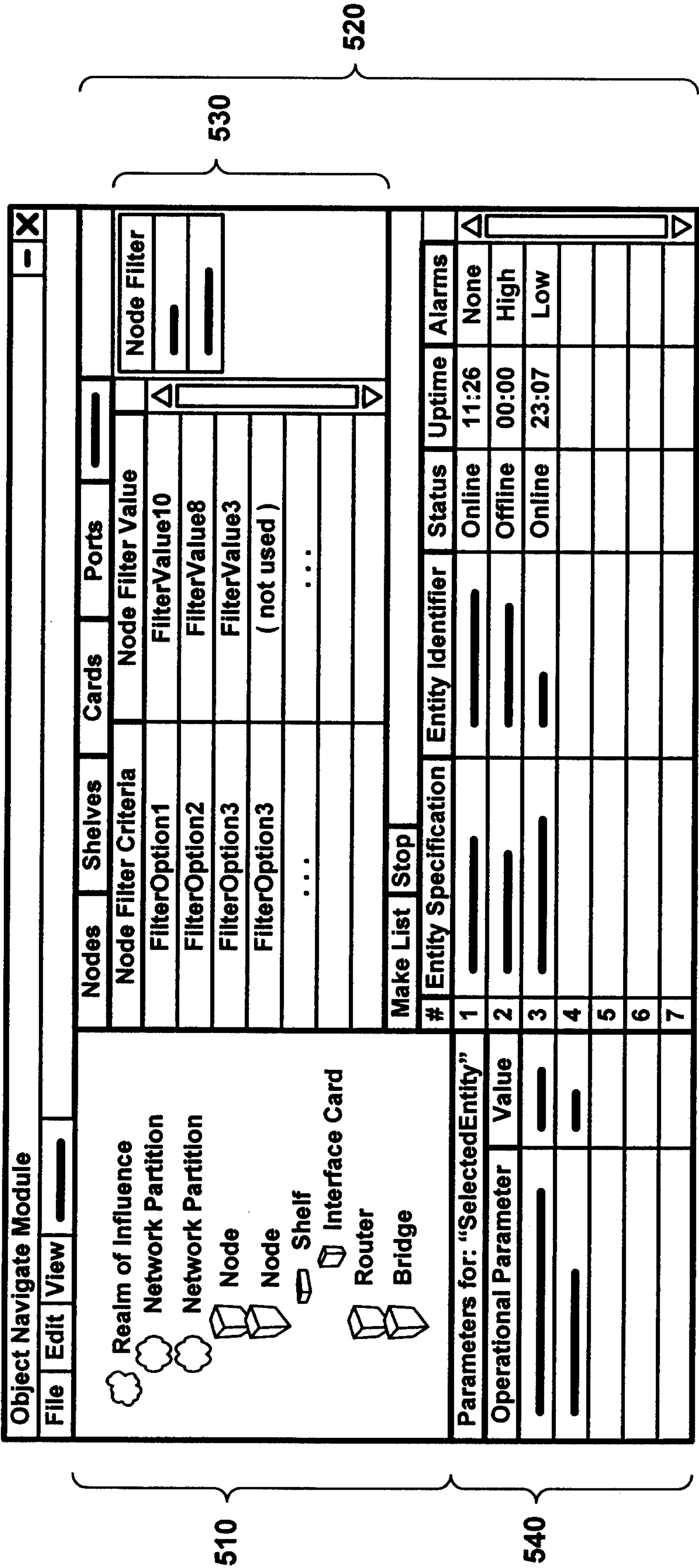


FIG. 2

Marko G. Clark



Marks & Clerk

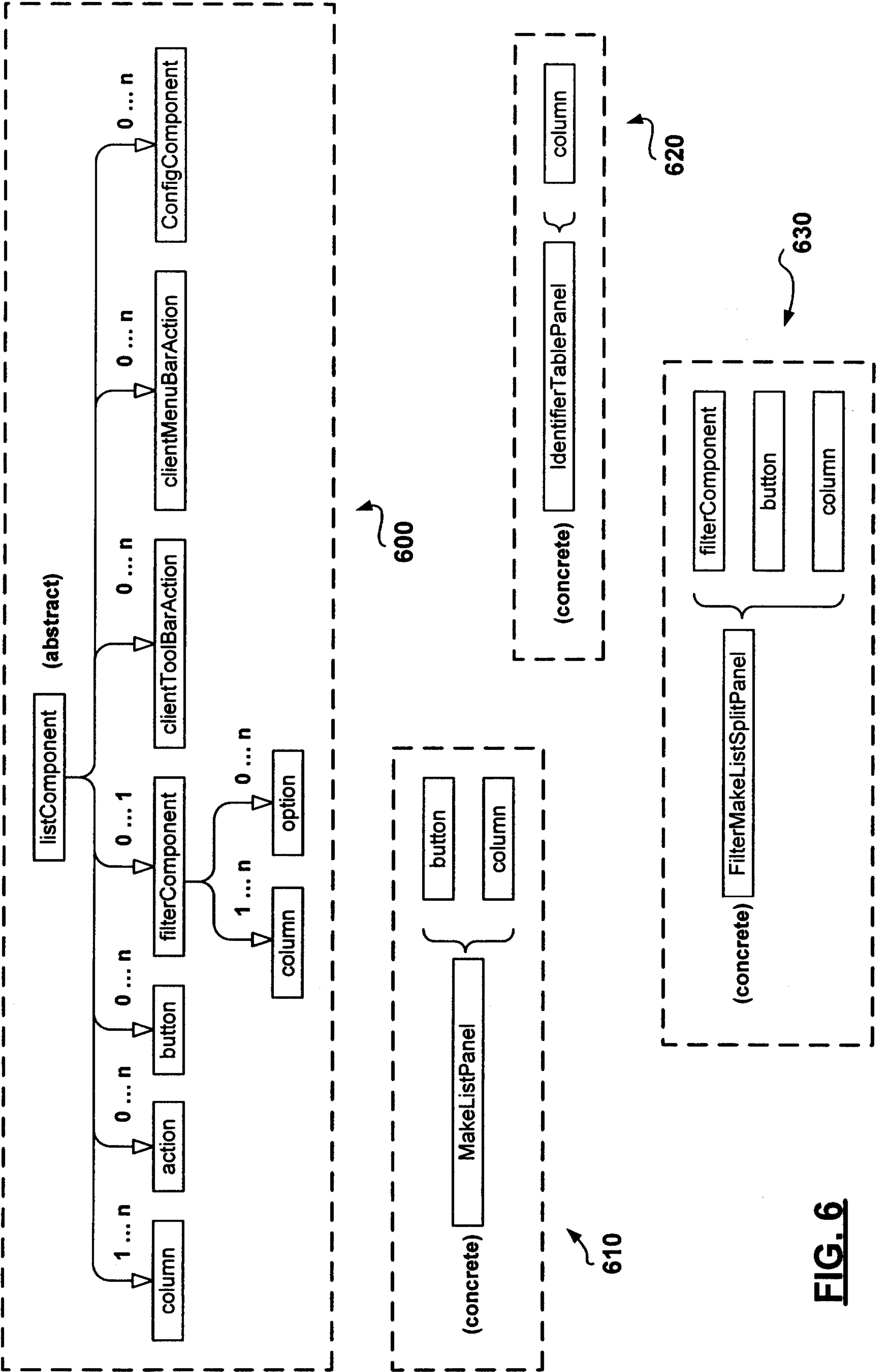


FIG. 6

Mark G. Blom

(EntityFilterMakeListSplitPanel)

Filter Criteria	Filter Value	Operation1	Operation2
FilterOption1	FilterValue10	AND	+
FilterOption2	FilterValue8	XOR	-
FilterOption3	FilterValue3	OR	-
FilterOption3	(not used)		
...	...	...	...

Make ListStop

#	Entity Specification	Entity Identifier	Status	Uptime	Alarms
1			Online	11:26	None
2			Offline	00:00	High
3			Online	23:07	Low
4					
5					

FIG. 7

700

(EntityConfigPanel)

EntityIdentifier

EntityType

NetworkAddress

ServiceAssociation

OK

Save

Cancel

Help

GeneralLayer-3Layer-2Layer-1

Parameter1

Parameter2

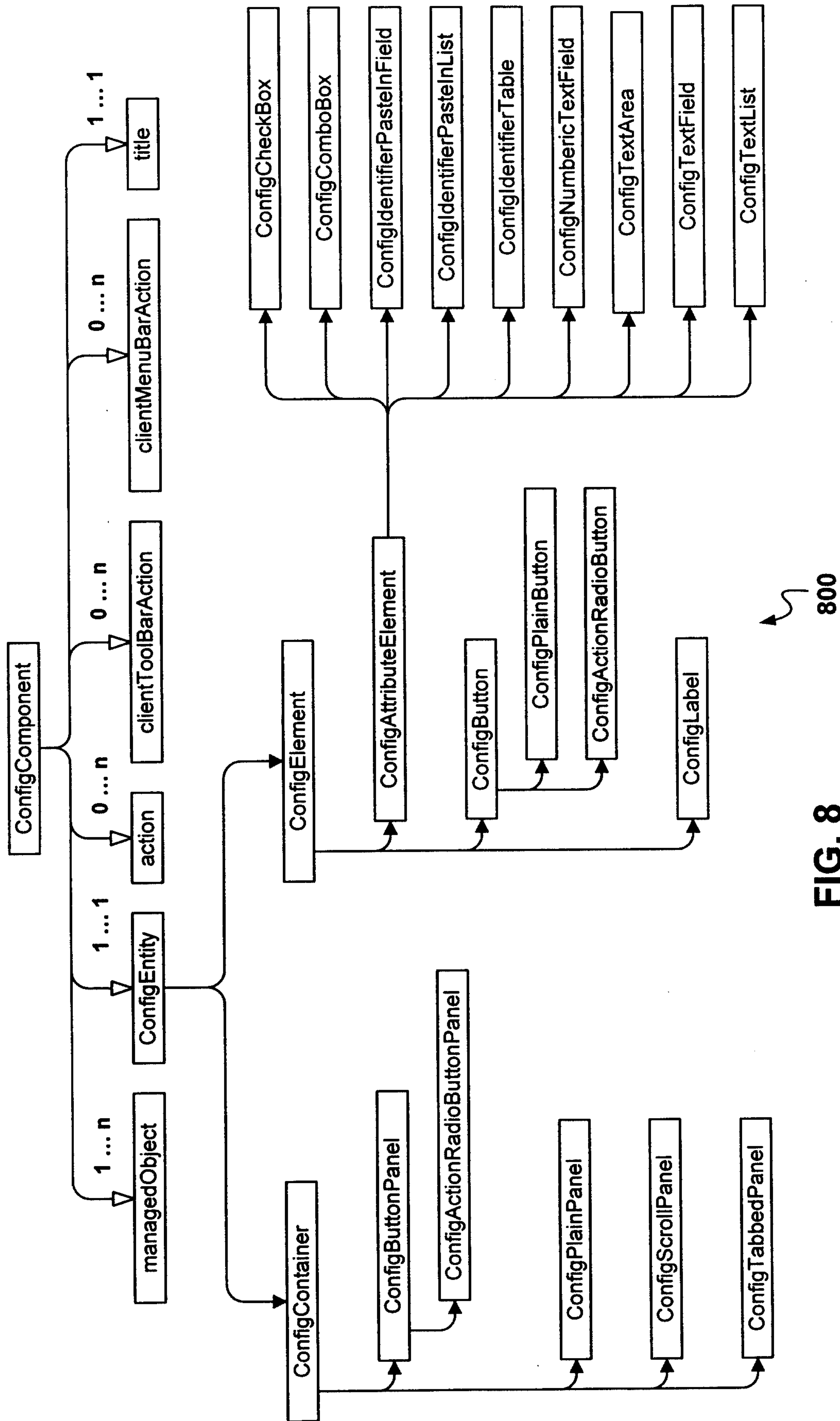
Commit Changes

Test

FIG. 9

900

Mario J. Clark

**FIG. 8**

Mark S. Clark

