

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
13 September 2007 (13.09.2007)

PCT

(10) International Publication Number
WO 2007/103722 A2

(51) International Patent Classification: **Not classified**

(21) International Application Number:

PCT/US2007/063061

(22) International Filing Date: 1 March 2007 (01.03.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/778,504 1 March 2006 (01.03.2006) US
11/680,289 28 February 2007 (28.02.2007) US

(71) Applicants (for all designated States except US): **SYMBOL TECHNOLOGIES, INC.** [US/US]; One Symbol Plaza, Holtsville, New York 11742 (US). **FOROOD, Natalie** [US/US]; San Diego, CA (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SANKA-NARAYANAM, Jayaram** [IN/US]; San Diego, CA (US). **FOWLER, David** [US/US]; San Diego, CA (US). **JEGHERS, Mark** [US/US]; 6482 Baguio Court, San Jose, CA 95119 (US).

(74) Agents: **INGRASSIA, Vincent B.** et al.; 7150 E. Camelback Rd., Suite 325, Scottsdale, Arizona 85251 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

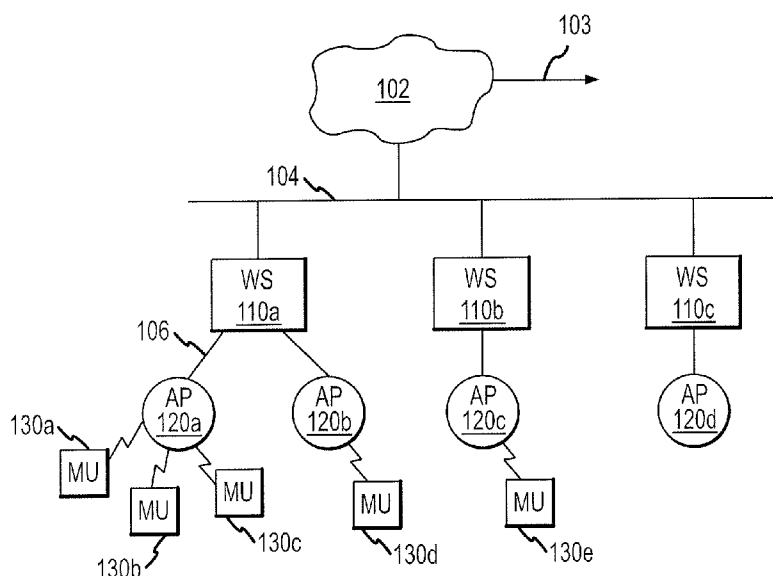
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: METHODS AND APPARATUS FOR THE CREATION OF APPLETS USED IN CONNECTION WITH USER INTERFACES



(57) Abstract: Applets are created for use within a user interface of the type configured to manage a wireless local area network (WLAN). A plurality of management information base (MIB) objects are provided, each associated with a corresponding plurality of components within the WLAN. A plurality of MIB XML files are also created, each of which is associated with a corresponding one of the plurality of MIB objects. A plurality of software interfaces, each of which is associated with a corresponding one of the plurality of MIB XML files, are generated. An applet (e.g., a Java applet) is constructed using a subset of the plurality of software interfaces.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND APPARATUS FOR THE CREATION OF APPLETS USED IN CONNECTION WITH USER INTERFACES

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Pat. Serial No. 60/778,504, filed March 1, 2006, and to U.S. Non-Provisional Pat. Serial No. 11/680,289 filed February 28, 2007.

TECHNICAL FIELD

[0002] The present invention generally relates to networked software and, more particularly, to generation of user interfaces based on the content of XML files and the conversion of MIB files to applets.

BACKGROUND

[0003] Due the size of modern wireless networks, it has become difficult to plan, monitor, manage, and troubleshoot the system as a whole. There has been a dramatic increase in demand for mobile connectivity solutions utilizing various wireless components and wireless local area networks (WLANs). Such WLANs generally include multiple wireless access points that communicate with mobile devices using one or more RF channels (e.g., in accordance with one or more of the IEEE 802.11 standards). The number of mobile units and associated access ports, as well as the number of RFID readers and associated antennae, can be very large in an enterprise. As the number of components increases, the management and configuration of those components becomes complicated and time-consuming.

[0004] Management software typically includes one or more user interfaces that incorporate management information base (MIB) entries and user interface applets. Currently known systems, however, are limited with respect to their ability to create user interfaces (e.g., Java interfaces) based on these MIBs.

[0005] Accordingly, it is desirable to provide improved methods of generating and controlling user interface structures employing applets. Other desirable features and characteristics of the present invention will become apparent from the subsequent detailed description and the appended claims, taken in conjunction with the accompanying drawings and the foregoing technical field and background.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] A more complete understanding of the present invention may be derived by referring to the detailed description and claims when considered in conjunction with the following figures, wherein like reference numbers refer to similar elements throughout the figures. The figures listed below are included in Attachment A, which is incorporated by reference below.

[0007] FIG. 1 depicts an exemplary WLAN environment useful in describing the present invention; and

[0008] FIG. 2 is a combination conceptual block diagram and flowchart that depicts an exemplary applet creation process.

DETAILED DESCRIPTION

[0009] The following detailed description is merely exemplary in nature and is not intended to limit the range of possible embodiments and applications. Furthermore, there is no intention to be bound by any theory presented in the preceding background or the following detailed description.

[0010] For simplicity and clarity of illustration, the drawing figures depict the general structure and/or manner of construction of the various embodiments. Descriptions and details of well-known features and techniques may be omitted to avoid unnecessarily obscuring other features. Elements in the drawings figures are not necessarily drawn to scale: the dimensions of some features may be exaggerated relative to other elements to assist improve understanding of the example embodiments.

[0011] Terms of enumeration such as “first,” “second,” “third,” and the like may be used for distinguishing between similar elements and not necessarily for describing a particular spatial or chronological order. These terms, so used, are interchangeable under appropriate circumstances. The embodiments of the invention described herein are, for example, capable of use in sequences other than those illustrated or otherwise described herein. Unless expressly stated otherwise, “connected” means that one element/node/feature is directly joined to (or directly communicates with) another element/node/feature, and not necessarily mechanically. Likewise, unless expressly stated otherwise, “coupled” means that one element/node/feature is directly or indirectly joined to (or directly or indirectly communicates with) another element/node/feature, and not necessarily mechanically.

[0012] The terms “comprise,” “include,” “have” and any variations thereof are used synonymously to denote non-exclusive inclusion. The terms “left,” “right,” “in,” “out,” “front,” “back,” “up,” “down,” and other such directional terms are used to describe relative positions, not necessarily absolute positions in space. The term “exemplary” is used in the sense of “example,” rather than “ideal.” For the purposes of conciseness, conventional techniques and systems related to data communication, network systems, wireless local area networks (WLANs), communication standards (e.g., the IEEE 802.11 family of specifications), and applet operation are not described in detail herein.

[0013] Although the present invention may be used in a variety of contexts – e.g., any system where applets or other applications are created and sent from one device to another device for execution – FIG. 1 depicts a wireless local area network (WLAN) useful in describing the present invention. In this embodiment, one or more switching devices 110 (alternatively referred to as “wireless switches,” “WS,” or simply “switches”) are coupled to a network 104 (e.g., an Ethernet network coupled to one or more other networks or devices, indicated by network cloud 102). One or more wireless access points 120 (alternatively referred to as “access ports” or “APs”) are configured to wirelessly connect to one or more mobile units 130 (alternatively referred to as “MUs”). The access points 120 are suitably connected to corresponding wireless switches 110 via communication lines 106 (e.g., conventional Ethernet lines). Any number of additional and/or intervening switches, routers, servers and other network components may also be present in the system. Additional servers and client systems (not shown) may communicate indirectly or directly with the various illustrated components (e.g., via a network connection 103).

[0014] A particular access point 120 may have a number of associated mobile units 130. For example, in the illustrated topology, mobile units 130(a), 130(b), and 130(c) are associated with access point 120(a), while mobile unit 130(e) is associated with access point 120(c). Furthermore, one or more access points 120 may be connected to a single wireless switch 110. Thus, as illustrated, access point 120(a) and access point 120(b) are connected to wireless switch 110(a), and access point 120(c) is connected to wireless switch 110(b).

[0015] Each wireless switch 110 determines the destination of data packets it receives over network 104 and routes that data packet to the appropriate access point 120 if the destination is a mobile unit 130 with which the access point 120 is associated. Each wireless switch 110 therefore maintains a routing list of mobile units 130 and their

associated access points 130. These lists are generated using a suitable packet handling process known in the art. Thus, each access point 120 acts primarily as a conduit, sending/receiving RF transmissions via mobile units 130, and sending/receiving data packets via a network protocol with wireless switch 110.

[0016] The access points 120 are typically capable of communicating with one or more mobile units 130 through multiple RF channels. This distribution of channels varies greatly by device, as well as country of operation. For example, in one embodiment in accordance with 802.11(b) there are fourteen overlapping, staggered channels, each centered 5 MHz apart in the RF band, although only eleven channels may currently be used legally in the U.S.

[0017] Referring to FIG. 2, an exemplary method of creating an applet in accordance with the present invention will now be described. Initially, a plurality of management information base (MIB) objects (“elements” or “files”) 204 are provided within a database 202. Each of these MIB objects 204 are associated with a corresponding plurality of components within the WLAN (e.g., the WLAN illustrated in FIG. 1), and in one embodiment are stored as static final Java objects.

[0018] As is known in the art, a management information base (MIB) is based on the OSI/ISO network model, and is a type of database used to manage devices in the network. It includes a set of objects (MIB objects) corresponding to the various switches, routers, and other components within the system. These objects are typically defined using a subset of Abstract Syntax Network Notion One (ASN.1), entitled “Structure of Management Information Version 2 (SMIV2),” RFC 2578. MIBs may be periodically updated to add new functionality, remove ambiguities, and to fix defects. These changes are preferably made in accordance with Section 10 of RFC 2578. Various tools exist to manipulate MIBs, including, for example, MIB2GUI and other MIB-related software tools offered by MG-Soft, Inc.

[0019] In accordance with the present invention, a plurality of MIB XML files 208 are created and stored within a database 206 (or within database 202). Each MIB XML file 208 is associated with a corresponding one of the plurality of MIB objects 204. The nature of XML (eXtensible Markup Language) is well known in the art, and need not be described herein. A translator (e.g., MG-Soft MIB2GUI) is preferably used to convert MIB objects to a form that supports a user interface (UI) puller/pusher infrastructure, and permits UI validation based on the MIB.

[0020] In one embodiment, when MIBs 204 are created or updated, an XML export file 208 is created using a suitable software (e.g., via MG-SOFT's MIB Explorer "Export to XML" feature). These XML export files are checked into a subversion repository within database 206. In one embodiment, this process is made part of the build so that system always has current XML files.

[0021] A plurality of software interfaces 210 are then created, each of which is associated with a corresponding one of the plurality of MIB XML files 208. These interfaces 210 may be stored prior to creation of the applet, or created at the same time that the applet assembled. In one embodiment, software interfaces 210 comprise Java interfaces that preferably comprise one or more constant classes corresponding to the corresponding SNMP objects. These constant classes might include, for example, an object ID (OID) of the SNMP object, the type of the SNMP object, and access information regarding the SNMP object. As is known in the art, SNMP ("Simple Network Management Protocol") is a widely-used protocol employed to manage elements in large networks, and is based on a manager/agent model. The manager provides the interface between the human network manager and the management system, and the agent provides the interface between the manager and the physical devices being managed. SNMP is powerful notwithstanding its relatively small command set. Further information regarding SNMP may be found, for example, within the IETF SNMPv1 and SNMPv2 standard documents.

[0022] Finally, an applet 212 is created using a subset of the plurality of software interfaces. As described above, for every MIB element 204 available to applet 212, there is a static final Java object which defines that MIB element, including its OID. These objects may be pre-created, one for each discrete MIB element, one for each table column, and one for each overall table. These objects are used to identify a MIB element 204 that the user wishes to manipulate.

[0023] The term "applet" is used to refer to any small software code, and is not limited to Java applets. In one embodiment, a management applet is created – i.e., an applet that is used in connection with the wireless switch or other component and which allows one or more components of the WLAN to be managed.

[0024] Creation of applet 212 may be performed on a conventional user computer system operated by a network manager or other individual, and may include various software and hardware components conventionally used in such systems. In one

embodiment, applet 212 includes an applet used in connection with a web browser to provide a user interface to one or more of the WLAN components.

[0025] When the management applet is built, the MIB XML files are converted into a set of matching Java interfaces using a MIB2GUI tool. Each interface contains one or more constant classes, where each class corresponds to an SNMP object. These classes contain, for example, the object's OID, type, optional range, and access info.

[0026] The following presents a sample constants file. To obtain the fully qualified SNMP OID value, the UI appends the constant's OID to the MIB_PREFIX. All constants are preferably preceded by "public static final." This prefix has been removed for clarity. Values are for illustration purposes only.

```
public interface SymbolSampleMib
{
    short[] MIB_PREFIX = new short[] {1,3,6,1,4,1,388};

    SNMPElement SNMP_SYMBOL =
        new SNMPElement(new short[] {}, DataType.STRING, null, true);
    SNMPElement SNMP_WSD =
        new SNMPElement(new short[] {11}, DataType.STRING, null, false);
    SNMPElement SNMP_WS2K =
        new SNMPElement(new short[] {11,2}, DataType.STRING, null, false);
    SNMPElement SNMP_CC_INFO =
        new SNMPElement(new short[] {11,2,2,1}, DataType.STRING, null, false);
    SNMPElement SNMP_CC_INFO_SERIAL_NUMBER =
        new SNMPElement(new short[] {11,2,2,1,1}, DataType.STRING, null, false);
    SNMPElement SNMP_CC_SNMP_ACCESS_V3_AUTH_ALGORITHM =
        new SNMPElement(new short[] {11,2,2,7,4,2,1,7}, DataType.INTEGER, null,
false);
    SNMPElement SNMP_CC_SNMP_ACCESS_V3_AUTH_PASSWORD =
        new SNMPElement(new short[] {11,2,2,7,4,2,1,8}, DataType.STRING, new
Range(8,31), false);

    // ccSnmAccessV12Table table start
    SNMPElement SNMP_CC_SNMP_ACCESS_V12_INDEX =
        new SNMPElement(new short[] {11,2,2,7,4,1,1,1}, DataType.INTEGER, new
Range(1,20), false);
    SNMPElement SNMP_CC_SNMP_ACCESS_V12_COMMUNITY =
        new SNMPElement(new short[] {11,2,2,7,4,1,1,2}, DataType.STRING, null,
false);
    SNMPElement SNMP_CC_SNMP_ACCESS_V12_CUSTOM_OID =
        new SNMPElement(new short[] {11,2,2,7,4,1,1,3}, DataType.OID, null,
false);

    SNMPTableElement(new short[] {11,2,2,7,4,1}, null, false,
        new SNMPElement[] {
            SNMP_CC_SNMP_ACCESS_V12_COMMUNITY,
            SNMP_CC_SNMP_ACCESS_V12_CUSTOM_OID
        });
}
```

[0027] The SNMPElement Class is given as:

```
com.symbol.pinnacle.applet.data.SNMPElement
```

[0028] and contains the following fields:

```
public class SNMPElement
{
    private short[] oidDigits = null;
    private DataType type = DataType.STRING;
    private Range range = null;
    private boolean readOnly = true;
}
```

[0029] OIDs may be stored as an array of short integers to reduce memory usage. Given the large number of constants that may be generated in any particular environment, storing the OID as an SNMP OID class (such as the ones provided in an SNMP4J or AdventNet library) consumes on the order of one additional megabyte of memory. In one embodiment, The DataType class is a type-safe enum class containing UI types mapped from SNMP types during the conversion process. “Range” contains a low and high limit for integer or string values. For strings, the low limit is the minimum length, and the high limit is the maximum length. The Boolean “readOnly” flag is TRUE if the field is read only, and FALSE if the object is read/write or write only.

[0030] The system preferably includes an SNMPTableElement Class. In one embodiment, the export tool (e.g., MIG2GUI) generates an SNMPTableElement class for each SNMP table. This class contains all the fields in SNMPElement, as well as an array of columns. Each column value in the SNMPTableElement is a previously-defined SNMPElement constant.

[0031] With respect to the make file, constants files may be updated by the translator when their corresponding source XML files 208 are modified. In addition, the translator is built when its source files are changed, or when SNMPElement and related classes are modified. Both targets are preferably implemented in a make file associated with the applet.

[0032] While at least one example embodiment has been presented in the foregoing detailed description, it should be appreciated that a vast number of variations exist. It should also be appreciated that the example embodiment or embodiments described herein are not intended to limit the scope, applicability, or configuration of the invention in any way. Rather, the foregoing detailed description will provide those skilled in the art with a convenient road map for implementing the described embodiment or embodiments. It should be understood that various changes can be made in the function and arrangement

of elements without departing from the scope of the invention and the legal equivalents thereof.

CLAIMS

1. A method of creating an applet for use within a user interface configured to manage a wireless local area network (WLAN) , the method comprising:

providing a plurality of management information base (MIB) objects associated with a corresponding plurality of components within the WLAN;

creating a plurality of MIB XML files, each of which is associated with a corresponding one of the plurality of MIB objects;

creating a plurality of software interfaces, each of which is associated with a corresponding one of the plurality of MIB XML files; and

creating the applet using a subset of the plurality of software interfaces.

2. The method of claim 1, wherein the software interfaces comprise Java interfaces, and the applet comprises a Java applet.

3. The method of claim 2, wherein each Java interface includes one or more constant classes, wherein each constant class corresponds to an SNMP object.

4. The method of claim 3, wherein each constant class includes an object ID (OID) of the SNMP object, the type of the SNMP object, and access information regarding the SNMP object.

5. The method of claim 2, wherein the step of creating a plurality of Java interfaces occurs substantially simultaneously with the step of creating the applet.

6. The method of claim 1, wherein the MIB objects are stored as static final Java objects.

7. A network management system comprising:

a management information base (MIB) database for storing: a plurality of management information base (MIB) objects associated with a corresponding plurality of components within the WLAN; a plurality of MIB XML files, each of which is associated with a corresponding one of the plurality of MIB objects; and a plurality of software interfaces, each of which is associated with a corresponding one of the plurality of MIB XML files; and

a user system configured to allow a user to create the applet using a subset of the plurality of software interfaces.

8. The system of claim 7, wherein the software interfaces comprise Java interfaces, and the applet comprises a Java applet.

9. The system of claim 8, wherein each Java interface includes one or more constant classes, wherein each constant class corresponds to an SNMP object.

10. The system of claim 9, wherein each constant class includes an object ID (OID) of the SNMP object, the type of the SNMP object, and access information regarding the SNMP object.

11. The system of claim 8, wherein the step of creating a plurality of Java interfaces occurs substantially simultaneously with the step of creating the applet.

12. The system of claim 7, wherein the MIB objects are stored as static final Java objects.

13. A wireless local area network (WLAN) comprising:

- a wireless switch configured to accept a management applet;

- a plurality of access ports communicatively coupled to the wireless switch;

- a plurality of mobile units, each communicatively coupled to at least one of the access ports; and

- a management information base (MIB) database for storing: a plurality of management information base (MIB) objects associated with a corresponding plurality of components within the WLAN; a plurality of MIB XML files, each of which is associated with a corresponding one of the plurality of MIB objects; and a plurality of software interfaces, each of which is associated with a corresponding one of the plurality of MIB XML files;

- wherein the applet comprises one or more of the plurality of software interfaces.

14. The network of claim 13, wherein the software interfaces comprise Java interfaces, and the applet comprises a Java applet.

15. The system of claim 14, wherein each Java interface includes one or more constant classes, wherein each constant class corresponds to an SNMP object.

16. The system of claim 15, wherein each constant class includes an object ID (OID) of the SNMP object, the type of the SNMP object, and access information regarding the SNMP object.

1/2

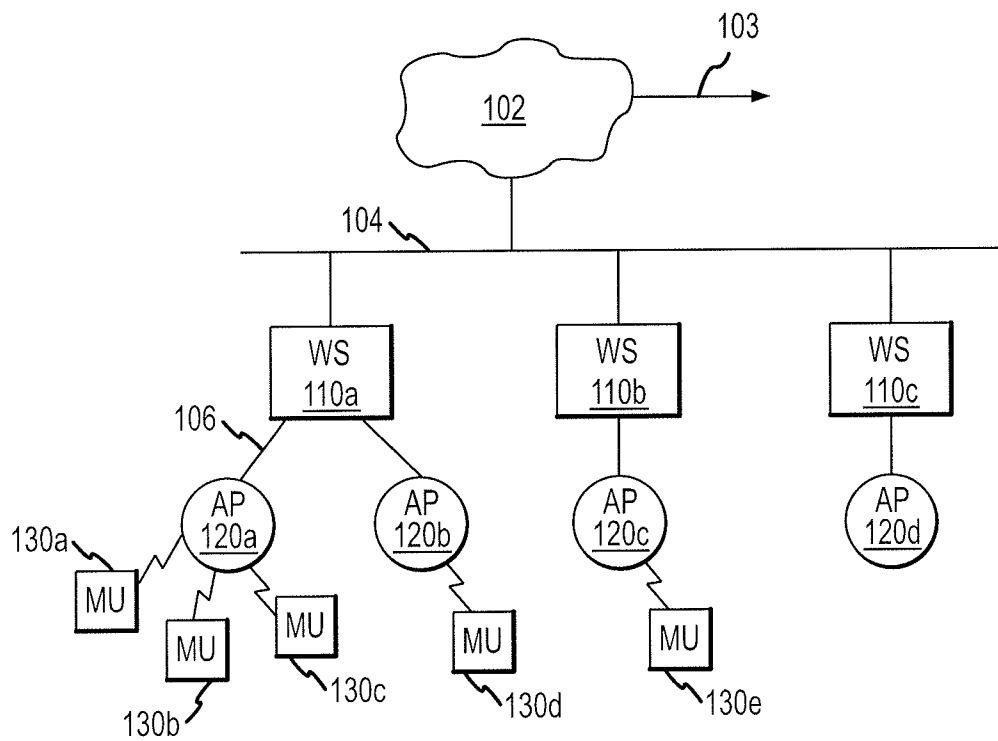


FIG.1

