

Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

VECTOR FLOW, INC.,
Petitioner

v.

HID GLOBAL CORPORATION,
Patent Owner

Case IPR2023-00353
U.S. Patent No. 8,234,704 B2
Issue Date: July 31, 2012

Title: PHYSICAL ACCESS CONTROL AND SECURITY MONITORING SYSTEM
UTILIZING A NORMALIZED DATA FORMAT

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,234,704 B2**

Table of Contents

	Page
I. MANDATORY NOTICES UNDER §42.8(A)(1)	8
A. Real Party-In-Interest under §42.8.(b)(1).....	8
B. Related Matters under §42.8(b)(2)	8
C. Lead and Back-Up Counsel under §42.8(b)(3)	8
D. Service Information	10
II. FEE PAYMENT	10
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D).....	10
A. Grounds for Standing	10
B. Identification of Challenge and Statement of Precise Relief Requested	10
C. Considerations Under §§ 314(a) and 325(d).....	11
IV. OVERVIEW OF THE PATENT.....	11
A. Level of Ordinary Skill in the Art	11
B. Specification Overview	11
V. CLAIM CONSTRUCTION	14
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	14
A. Overview of Grounds	14
B. Ground 1: Claims 11-14 Are Obvious Over Farino, Richman, and Wiegel.....	16
1. Claim 11	16
(a) “interfacing in a centralized security system, a plurality of security sensor types distributed throughout a plurality of sites, each sensor type configured to respond to a corresponding type of actionable event, each sensor type provided by a different manufacturer of a plurality of manufacturers;” (Claim 11[a]).....	16

Table of Contents
(continued)

Page

(b)	“accepting sensor data from each security sensor [sic] an integration module including an agent for each type of security sensor, wherein the sensor data from each security sensor is embodied in a native data representation format of each respective manufacturer of the plurality of manufacturers;” (Claim 11[b])	23
(c)	“defining individual user profiles and their respective access privileges and credentials in the system;” (Claim 11[c])	29
(d)	“mapping the sensor data from each security sensor in the native data representation format of each manufacturer of the plurality of manufacturers to a common data representation format, the common data representation format including a data object and processing information for the sensor data;” (Claim 11[d]).....	31
(e)	“generating unique physical access privileges and credentials to exclusively map a defined user profile to a spatial hierarchy of physical sites along with security devices of the system, wherein the unique physical access credentials maintain a common representation of the user’s identity across the plurality of sites and to associate specific user identities with respective actionable events;” (Claim 11[e])	35

Table of Contents
(continued)

Page

(f)	“defining physical security policies of the site in the context of user profiles at all sites through actionable representations of physical, network and information technology resources of the site, wherein the security policies define standardized rule definitions through visual rules depicted by live objects that contain attributes to define their spatial relationship to the actionable representations, and that are applied to the actionable events normalized to the common data representation format to produce normalized event data; and” (Claim 11[f])	42
(g)	“receiving the normalized event data and applying relevant transformation and routing rules comprising condition-action sequences in order to maintain user profiles and physical security states across the plurality of sites and to resolve the actionable events through the associated specific user identities.” (Claim 11[g])	59
2.	Claim 12: The method of claim 11 wherein the plurality of sensor types comprise physical access, intrusion, and environmental conditions, and wherein the action events comprise at least one of generating an audible alarm, notifying authority personnel, initiating lockdown mode of the site, initiating computer backup and shutdown procedures, and facilitating exit from the site.”	63
3.	Claim 13: “The method of claim 11 further comprising converting the native data representation format for each security sensor into XML (Extensible Markup Language) format, and wherein the native data representation format encompasses native sensor data including text data and binary data represented in a proprietary format, and wherein the native sensor data includes at least one of a command or sensor state information.”	64

Table of Contents
(continued)

	Page
4. Claim 14: “The method of claim 11 further comprising applying, through a rules engine, defined rules on the data and commands from each security sensor, wherein each rule has an associated action and one rule is applied at a time to the normalized event data in a sequential order dictated by one of a defined rule execution order or a top to bottom order based on event time.”	68
C. Ground 2: Claim 15 is Obvious Over Farino, Richman, Wiegel, and Moore.....	70
1. Claim 15: “The method of claim 11 further comprising representing, through a visual editor, each normalized physical sensor as an object containing one or more attributes that define a spatial relationship to other objects through one or more defined rules, the visual editor including a drag and drop graphical user interface component that facilitates creation of rules through interconnection of rule objects and definition of rule properties.”	70
VII. CONCLUSION.....	81

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 8,234,704 B2 to Vikrant Ghai et al. (filed April 17, 2007, issued July 31, 2012) (“’704” or “’704 patent”)
1002	Declaration of Vijay K. Madisetti, Ph.D. (“Madisetti”)
1003	U.S. Patent Application Pub. No. 2007/0094716 A1 to Mark William Farino et al. (filed October 26, 2005, published April 26, 2007) (“Farino”)
1004	U.S. Patent Application Pub. No. 2003/0206100 A1 to Lawrence Richman et al. (filed June 20, 2002, published November 6, 2003) (“Richman”)
1005	U.S. Patent No. 6,484,261 B1 to Scott L. Wiegel (filed December 11, 1998, issued November 19, 2002) (“Wiegel”)
1006	U.S. Patent Application Pub. No. 2005/0246352 A1 to Jason F. Moore et al. (filed April 30, 2004, issued November 3, 2005) (“Moore”)
1007	U.S. Patent Application Pub. No. 2003/0023874 A1 to Rudy Prokupets et al. (filed July 16, 2001, published January 30, 2003) (“Prokupets”)
1008	U.S. Patent Application Pub. No. 2004/0267623 A1 to John H. Vivadelli et al. (filed January 9, 2003, published December 30, 2004) (“Vivadelli”)
1009	U.S. Patent Application Pub. No. 2002/0174357 A1 to Michael Davis et al. (filed April 6, 2001, published November 21, 2002) (“Davis”)
1010	Excerpts from <i>Microsoft Computer Dictionary</i> (5th ed. 2002)
1011	Declaration of Mark Jones in Support of Plaintiff HID Global Corporation’s Opening Claim Construction Brief, served in <i>HID Global Corp. v. Vector Flow, Inc. et al.</i> , No. 21-1769 (GBW)

Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

List of Exhibits

Exhibit No.	Description of Document
1012	Declaration of Stephen Gray in Support of Defendant Vector Flow's Claim Construction Brief, served in served in <i>HID Global Corp. v. Vector Flow, Inc. et al.</i> , No. 21-1769 (GBW)

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest under §42.8.(b)(1)

Vector Flow, Inc. is the real party-in-interest to this IPR petition.

B. Related Matters under §42.8(b)(2)

The '704 patent is the subject of pending litigation involving Petitioner: *HID Global Corp. v. Vector Flow, Inc.*, No. 21-1769 (GBW) (D. Del.). Petitioner was first served on December 17, 2021. In the underlying litigation, Patent Owner contends that Petitioner is barred from challenging the validity of the '704 patent under the equitable doctrine of assignor estoppel because the '704 patent inventors are now affiliated with Petitioner. Petitioner disputes application of assignor estoppel and the district court has issued no ruling on the issue. Nevertheless, Patent Owner's contentions regarding assignor estoppel have no applicability to IPR. *See Arista Networks, Inc. v. Cisco Sys., Inc.*, 908 F.3d 792, 804 (Fed. Cir. 2018) ("[A]ssignor estoppel has no place in IPR proceedings.").

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700	Andrew C. Mace (Reg. No. 63,342) amace@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW

Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

LEAD COUNSEL	BACK-UP COUNSEL
Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Suite 700 Washington, D.C. 20004 Tel: (650) 843-5808 Fax: (650) 849-7400
	Lowell D. Mead (Admission <i>pro hac vice</i> to be requested) lmead@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington, D.C. 20004 Tel: (650) 843-5734 Fax: (650) 849-7400
	Mark R. Weinstein (Admission <i>pro hac vice</i> to be requested) mweinstein@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington, D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400
	David N. Murdter (Admission <i>pro hac vice</i> to be requested) dmurdter@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington, D.C. 20004 Tel: (650) 843-5881 Fax: (650) 849-7400

D. Service Information

This Petition is being served by Priority Mail Express to the attorney of record for the '704 patent, 111078 - Staniford Tomita LLP, P.O. Box 591838, San Francisco, CA 94159. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

II. FEE PAYMENT

Petitioner requests review of 5 claims, with a \$41,500 payment.

**III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND
CONSIDERATIONS UNDER §§ 314(A) AND 325(D)**

A. Grounds for Standing

Petitioner certifies that the '704 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

**B. Identification of Challenge and Statement of Precise Relief
Requested**

Petitioner requests IPR institution based on:

Ground	Claims	Basis for Challenge under §103
1	11-14	Obvious over Farino (EX1003), in view of Richman (EX1004) and Wiegel (EX1005)
2	15	Obvious over Farino, in view of Richman, Wiegel and Moore (EX1006)

Submitted with this Petition is the Declaration of Vijay K. Madiseti (EX1002) ("Madiseti"), a qualified technical expert. (Madiseti, ¶¶1-12, Ex. A.)

C. Considerations Under §§ 314(a) and 325(d)

§314(a): The *General Plastic* factors are not relevant as this is the only IPR petition against the '704 patent. Nor do the *Fintiv* factors support discretionary denial under §314(a). The pending litigation against Petitioner is in early stages, and no claim construction rulings and no significant discovery have occurred.

§325(d): None of the references in the grounds listed in **Part III.B** above was cited during prosecution of the '704 patent.

IV. OVERVIEW OF THE PATENT

A. Level of Ordinary Skill in the Art

A person of ordinary skill in the art would have possessed a bachelor's degree in software engineering, computer science, computer engineering, or electrical engineering with at least two years' experience in distributed or network-based computer systems, and would have had a working knowledge about various ways to receive data from network-connected devices (such as sensors) and then process and/or act upon that data as needed. A person could also have qualified with more education and less technical experience, or vice versa. (Madisetti, ¶¶15-16, 18.)

B. Specification Overview

The '704 patent, entitled "Physical Access Control and Security Monitoring System Utilizing a Normalized Data Format," states that it "relate[s] generally to physical security systems, and more specifically to a system for integrating disparate

security systems using a rules-based policy engine and normalized data format.”

(’704, 1:24-27.) Figure 1A below illustrates such a system:

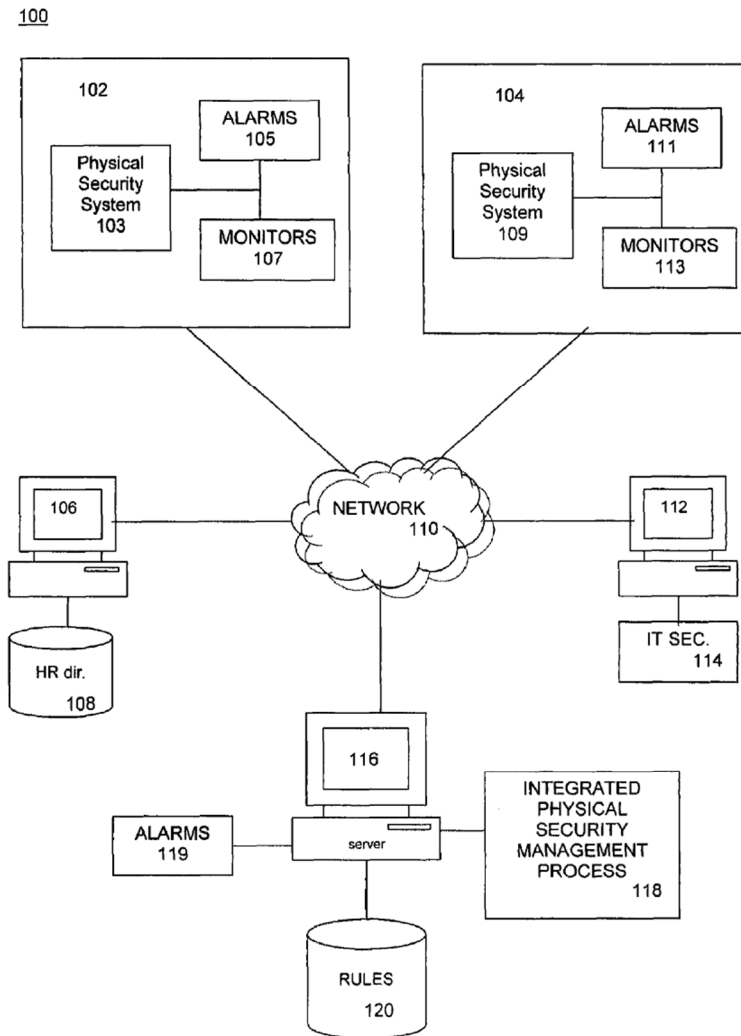


FIG. 1A

(’704, Fig. 1A.) As shown, computer network system **100** includes two facilities, **102** and **104** (such as a house, office building or other type of structure that contains some level of physical security infrastructure), each including physical security systems and associated security elements (such as access control, monitors and

sensors). ('704, 4:3-15, 4:40-45.)

Because the physical security systems at each facility may be provided by different vendors, the systems may provide data that is unique or proprietary. ('704, 7:42-48.) The computer network system **100** also includes server **116**, which executes an integrated physical security management process **118**. ('704, 4:58-64.) For example, process **118** includes “a normalization process that normalizes of [sic] disparate security systems data to a standard data format, for example, access control and alarm data.” ('704, 5:1-5; *see also id.*, 7:37-48.)

Process **118** also includes a management function that “provides a mode of visual representation of the normalized physical security systems, data and processes, and visual policy objects define the design time behavior for flexible and actionable rule creation.” ('704, 6:51-58.) Figure 7 below shows a visual policy editor that facilitates creation of security policies and related rules:

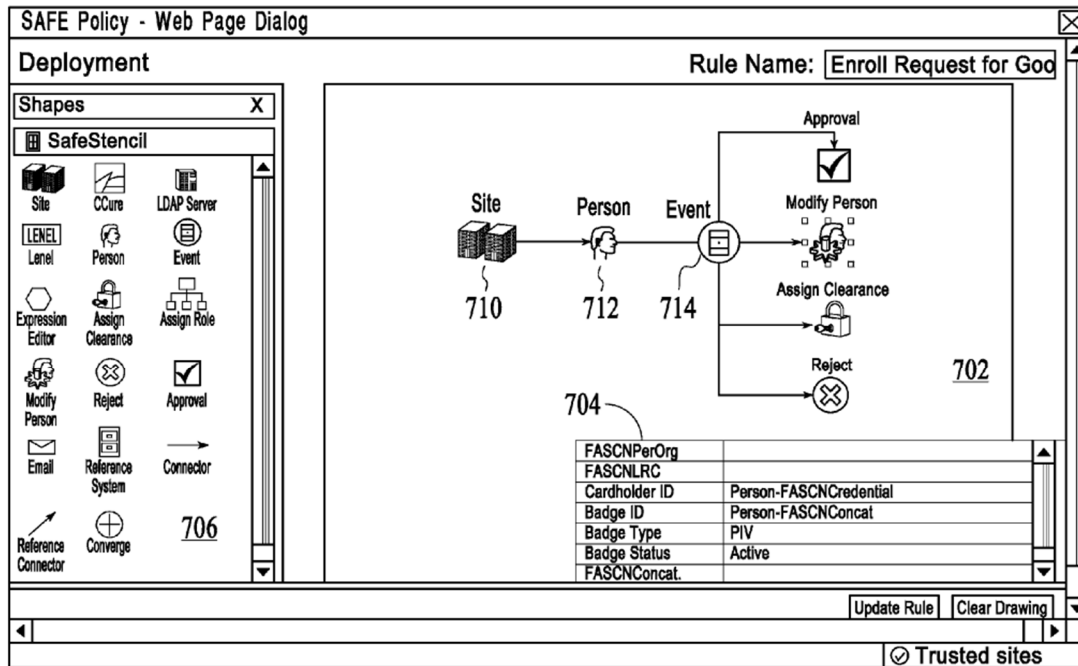


FIG.7 700

(’704, Fig. 7; *see also id.*, 2:53-56, 5:24-30, 15:32-47.) With the visual policy editor, standard rules and actions can be created and then applied across disparate systems. (’704, 5:26-35, 5:43-45, 7:48-52, 9:13-15.)

V. CLAIM CONSTRUCTION

Petitioner does not believe express claim construction is necessary at this time for this IPR. The parties in the pending litigation have exchanged constructions for certain terms, but the Petition discloses the claim limitations under those constructions or any other reasonable construction. (Madisetti, ¶¶26, 40.)

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Grounds

The challenged claims recite a security system in which sensor data is received from security sensors in a “**native data representation format**,” such as a vendor-

specific proprietary format, and converts it into a “**common data representation format**” for use by the system. The challenged claims further recite aspects of a graphical user interface for creating security policies.

The Petition cites **Farino** (EX1003) as the primary reference. It discloses a centralized security system (unified access control server) that receives and processes sensor data from a variety of security sensors such as card/badge readers. (Madisetti, ¶¶44-45.) The Petition combines Farino with **Richman** (EX1004) which further discloses converting vendor-specific sensor data into standard XML format. (Madisetti, ¶¶48-51.) With respect to the claimed user interface features, the Petition further combines with **Wiegel** (EX1005), which discloses a graphical user interface for creating security policies. (Madisetti, ¶¶53-57.) The combination of Farino, Richman and Wiegel renders obvious claims 11-14, as shown below. The Petition also cites **Moore** (EX1006), for **Ground 2** addressing dependent claim 15.

These prior art references qualify as prior art under at least 35 U.S.C. § 102(e) (pre-AIA) because they were filed before the earliest priority date of the ’704 patent. (Madisetti, ¶¶43, 47, 52, 58.) Petitioner has also cited additional references (EX1007-1010) for purposes of establishing basic knowledge of an ordinarily skilled artisan rather than to supply a claim limitation, and these also pre-date the ’704 patent. (Madisetti, ¶42.)

B. Ground 1: Claims 11-14 Are Obvious Over Farino, Richman, and Wiegel

1. Claim 11

The preamble recites “[a] **method comprising.**” The prior art renders obvious the claimed method as explained below.

- (a) **“interfacing in a centralized security system, a plurality of security sensor types distributed throughout a plurality of sites, each sensor type configured to respond to a corresponding type of actionable event, each sensor type provided by a different manufacturer of a plurality of manufacturers;” (Claim 11[a])**

Farino discloses a **“centralized security system”** in the form of a system that includes a unified access control server:

FIG. 6 illustrates one embodiment of the present invention where a unified access control server 200 for network and physical premises access is shown. Server 200 interfaces with both the legacy physical security system 202 and the frame/packet network 204 to control a unified physical access and/or network access. This system may be adapted to monitor and control access to buildings, homes, physical infrastructure as well as to information and network systems.

(Farino, ¶0061; *see also* Farino, Fig. 6, 7A-7C (unified access control server 200), Fig. 10 (unified access control server 400), Fig. 8 & ¶0085 (showing unified access control server integrated into network infrastructure device 300 such as a router, switch or access point).) Farino thus discloses a **“security system.”**

With respect to a **“centralized”** security system, Figure 10 shows unified

access control server **400** in a centralized configuration:

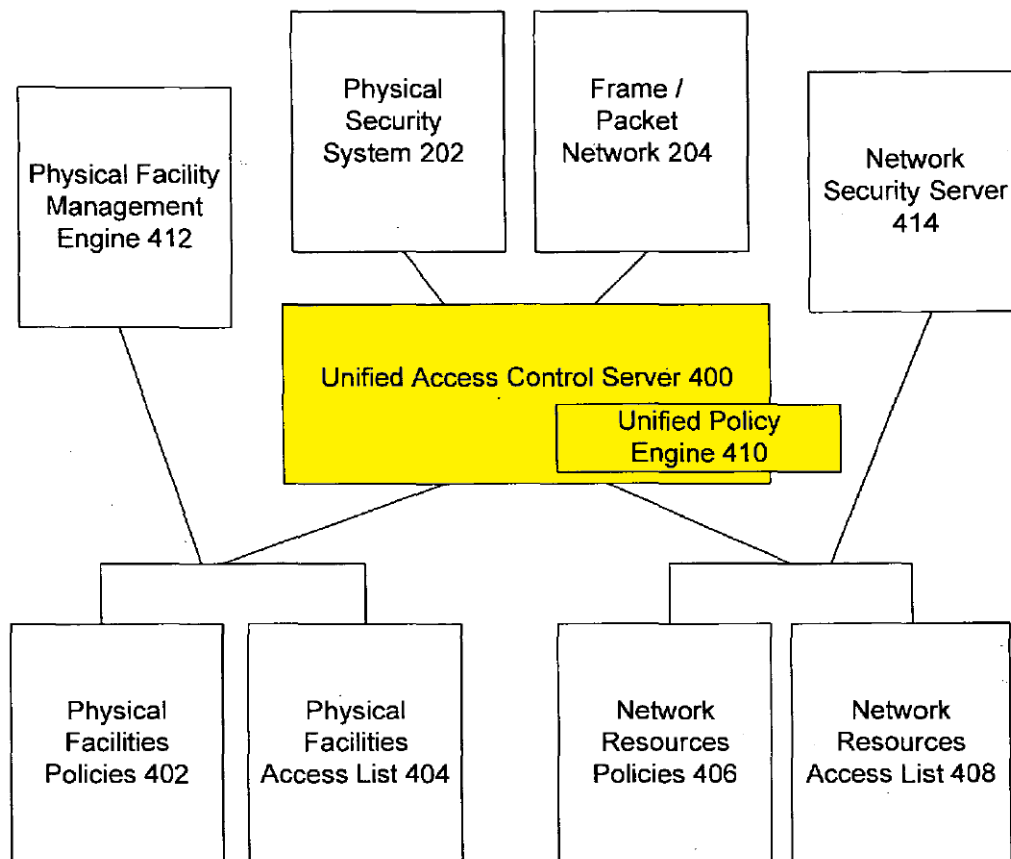


FIGURE 10

(Farino, Fig. 10 (highlighting added).) Farino explains that the access control server “may either hold all or a portion of the policies, rules and authorized users in a centralized or distributed fashion.” (Farino, ¶0027.) Farino further states that “[t]he unified access control server implements access policy that regulates user access to multiple facilities.” (Farino, ¶0083.) This allows the unified access control server to provide “unified access to both physical and network access logs [and] enhances both physical and network security.” (Farino, ¶0069.)

Farino thus provides a “**centralized**” security system, as claimed, because it provides a unified (*i.e.*, centralized) system for storing access control information and enforcing security policies across the one or more sites managed by the system. (Madisetti, ¶64.) As explained for 11[c] and 11[e] below, unified access control server also provides a centralized information repository (*e.g.* lists or tables) for storing access privilege, credential, and security policy information across the system. (Farino, ¶¶0097, 0084, 0069, 0062.)

Farino further discloses “**interfacing in a centralized security system, a plurality of security sensor types distributed throughout a plurality of sites.**” The “**plurality of security sensor types**” in Farino includes access control devices (ACDs) such as card readers, cameras, alarms, among others:

The present invention assumes that one or more access control devices (ACD) **110** and computers **151** or other network edge devices are part of a facility and its network, respectively. Common access control devices include card readers, biometric sensors, cameras, alarms, motion sensors and electro-mechanical door locks.

(Farino, ¶0063; *see also id.*, ¶0014.) The listing above provides mere examples. Farino discloses other **sensor types** including “fire and smoke sensors” (¶0079).

Farino also discloses that the plurality of security sensor types may be “**distributed throughout a plurality of sites,**” *i.e.*, throughout multiple buildings and/or facilities. (Farino, *e.g.*, ¶¶0075 (“Further, it should be noted that the unified

system may be deployed in one building or may include multiple facilities each of which is configured with network connected ACDs, legacy devices or a combination of network connected ACDs and legacy devices.”), 0083 (“The unified access control server implements access policy that regulates user access to multiple facilities.”); Madisetti, ¶66.)

Farino also discloses “***interfacing* in a centralized security system**” the plurality of sensor types. This is because security sensors can communicate with the unified access control server, for example, in response to a request for access. (Farino, *e.g.*, ¶0090 (“A user presents at a reader or other ACD. The detected request and associated credential information is transmitted as an access request message in one or more packets to the unified access control server **200.**”).) Moreover, unified access control server interfaces with each of the sensor types to provide a central hub for responding to sensor input. (Farino, *e.g.*, ¶¶0064-0065; Madisetti, ¶67.) In one embodiment, the sensor types in Farino interface with the centralized security system when the unified access control server receives information from a door control panel (DCP) that receives sensor data from an ACD. (Farino, *e.g.*, ¶¶0063 (“Each ACD **110** is coupled to door control panel (DCP) **119** that, in turn, is coupled to a packet/frame-enabled access control gateway **125**... The DCP access request messages are received and responded to by the unified access control server **200.**”), 0074-0075.) As a further disclosure of the “***interfacing***” step, network-connected

ACDs are registered with the unified access control server. (Farino, ¶¶0077; Madisetti, ¶¶67-68.)

Farino also discloses that “**each sensor type [is] configured to respond to a corresponding type of actionable event,**” because each type of sensor is designed to respond to the particular type of event it was designed to detect. (Madisetti, ¶69.) For example, for ACDs that govern physical access, an actionable event may correspond to a user presenting credentials (such as a security badge) to request physical access. (Farino, ¶¶0089-0090.) The “**actionable event**” there corresponds to an event in which a user attempts to present credentials, which can occur in response to the user presenting a security badge, providing fingerprints or a password, or other means depending on the type of sensor. (Farino, ¶¶0082, 0087, 0004-0005.) To take a few examples, the actionable event for a card reader occurs in response to presenting an encoded plastic card/security badge; for a biometric ACD, in response to presenting fingerprints or a retinal scan; and for a keypad ACD, in response to entering an appropriate code. (Farino, *e.g.*, ¶¶0005, 0063, 0068, 0089-0090, 0097.) As noted, Farino discloses other types of sensors including motion sensors and fire/smoke detectors, and it would have been obvious that these sensors are also configured to respond to corresponding actionable events (*e.g.* detection of motion or fire/smoke). (Madisetti, ¶69; Farino, *e.g.*, ¶0047.) Each of the sensor types in Farino qualifies as a type of “**sensor**” because it responds to a physical

stimulus, such as information from a security badge, biometric information from the user, input to a keypad, environmental conditions (*e.g.*, smoke or fire), and provides a resulting signal. (Madisetti, ¶69.)

Farino also renders obvious that “**each sensor type [is] provided by a different manufacturer of a plurality of manufacturers.**” It would have been obvious that each sensor type in Farino could have been provided by a different manufacturer or vendor. (Madisetti, ¶¶70-71.) Farino explains that each vendor of a security sensor may have a vendor-specific way of communicating to other access control products. (Farino, ¶¶0006 (“In many cases, the signaling protocols may be specific to a single vendor’s access control products.”), 0009 (“The actual communications between physical access control components is typically proprietary to a single vendor...”).) Farino addresses this issue by disclosing security access device gateways 125 that “can translate various access control system device vendor’s signaling formats and protocols into another access control system vendor’s component formats.” (Farino, ¶0040; *see also id.*, ¶0063.) Farino thus contemplates sensor types from different manufacturers. (Madisetti, ¶70.)

More fundamentally, an ordinarily skilled artisan would have appreciated that market forces, competition and specialization would have resulted in different companies manufacturing and selling specific sensor types. (Madisetti, ¶71.) For example, Prokupets (EX1007), cited merely to confirm the knowledge of an

ordinarily skilled artisan, identifies several examples of vendor-specific specialization of sensor types. (Madisetti, ¶71.) Prokupets identifies, for example, “access control systems” manufactured by “ONGUARD® systems sold by Lenel Systems International, Inc., Rochester, N.Y.” (¶0002), an “intrusion detection system” manufactured by “Digital Monitoring Products, Inc of Springfield, Mo.” (¶0023), a “fire system” which may be “manufactured by Pyrotronics, Inc. of Cedar Knolls, N.Y.” (¶0024). It thus would have been obvious that the sensor types in Farino could have come from a variety of manufacturers, such as (1) security sensor types including door access controls from a first manufacturer like Lenel Systems, (2) motion detection/intrusion sensors provided by a second manufacturer like Digital Monitoring Products, as identified in Prokupets, etc. (Madisetti, ¶71.)

The sensor manufacturers in Prokupets are mere examples. It would have been obvious for a security system to have had plurality of sensor types, each having been provided by a different manufacturer. This would have been a natural result of specialization within the industry (*e.g.* each manufacturer specializes in a particular sensor type). An ordinarily skilled artisan would have been motivated to construct a security system for an organization in such a manner, among other things (1) to leverage the specialization of individual manufacturers with respect to particular sensor types, and (2) to avoid the “single sourcing” risks associated with obtaining multiple sensor types from a single manufacturer. (Madisetti, ¶71.)

- (b) “accepting sensor data from each security sensor [sic] an integration module including an agent for each type of security sensor, wherein the sensor data from each security sensor is embodied in a native data representation format of each respective manufacturer of the plurality of manufacturers;” (Claim 11[b])

Petitioner notes that the claim recites “accepting sensor data... an integration module,” which appears to be a typographical error that omits a word before “an integration module.” Patent Owner and Petitioner have interpreted this language to mean “accepting sensor data... by an integration module,” which Petitioner has applied here.

Farino discloses “accepting sensor data from each security sensor,” which corresponds to data received from the security sensors. In the example of ACDs governing physical access, as noted, the sensor data reflects credential information provided by the user. (Farino, *e.g.*, ¶0090.) The “sensor data” there corresponds to information from a security card, fingerprint information, password, among others. (Farino, ¶¶0005, 0097.) It would have been obvious that all of the sensor types of Farino would have provided sensor data. (Madisetti, ¶73.)

Farino further discloses accepting this sensor data by “an integration module.” Claim 11 is a method claim and neither Patent Owner nor Petitioner have taken the position that “integration module” is a means-plus-function term. The integration module under the proposed combination takes the form of hardware and

software for receiving and processing sensor data from the security sensors, and is disclosed by the combination of Farino with **Richman**.

With respect to Farino, it discloses an access control gateway **125** that can translate information provided by various security access devices (including from “legacy” vendors), each having their own signaling formats and protocols. (Farino, ¶¶0040, 0063.) These gateways “can translate various access control system device vendor’s signaling formats and protocols to another access control system vendor’s component formats.” (Farino, ¶0040.) The access control gateway in Farino, in other words, is responsible for accepting sensor data from the sensor types and **integrating** it into the system by translating it into another format for use by the unified access control server. (Farino, ¶0063.) But Farino does not describe the operation of the access control gateway **125** in detail, and as such, Petitioner has relied on Farino in further combination with Richman (**EX1004**).

Richman discloses a security system similar to Farino that can receive sensor data from multiple types of security devices, such as video cameras, motion sensors, among others. (Richman, ¶¶0079, 0080.) Similar to Farino, Richman explains that these devices transmit information using varying device languages. (Richman, *e.g.*, ¶0092 (“[S]ecurity devices and sensors transmit data in device language specific for that device or sensor.”).) Richman responds by disclosing a system security core application (SCA) that “translates the device language into standardized converted

messages before input into and use by the SCA.” (Richman, ¶0065.) More specifically, Richman discloses **an integration module** in the form of a series of “conversion modules” that perform a function similar to the access control gateway **125** in Farino by receiving sensor data in its native form translating it into standardized data in XML format:

Conversion Module **202**. This module receives data from security hardware devices of varying types in their own specialized unique data format and converts this data into a standardized XML formatted message. Each unique hardware device requires a separate customized conversion module to translate its data into the coded SCA XML format for the purpose of further analysis. Thus, for this module, the input is a varying hardware signal, and the output is a standardized message in XML format.

(Richman, ¶0097.) XML refers to eXtensible Markup Language, a well-known and widely used language for encoding and exchanging information between systems and applications. (Madisetti, ¶76; Richman, ¶¶0030-0041.) “XML is a low-level syntax for representing structured data. You can use this simple syntax to support a wide variety of applications.” (Richman, ¶0043.) As Petitioner will explain in more detail for claim 11[d] below, Richman provides several concrete examples of converting device-specific data from security devices into data in standardized XML format. (Richman, ¶¶0120-0124 (tenant seeks to access room secured with a keypad), 0128-0129 (tenant seeks access to a room secured with a fingerprint or eye

detection system), 0117-0119 (motion sensor detects that a window has been broken); Madisetti, ¶¶77-78, 90-91.)

Manner, Rationale and Motivation to Combine: It would have been obvious to combine Farino with Richman, predictably resulting in the security system of Farino in which the access control gateway **125** (Farino, ¶¶0040, 0063), further incorporates the XML conversion techniques of Richman. (Madisetti, ¶79.) Under the proposed combination, the system would have “**accept[ed] sensor data from each security sensor [by] an integration module,**” *i.e.*, sensor data would have been accepted by an access control gateway **125** in Farino, which would have been further adapted to include conversion module functionality of Richman. Farino and Richman are analogous references in the same field and would have been reasonably pertinent to a number of problems facing the inventors. (Madisetti, ¶82.)

The integration module under the proposed Farino-Richman combination would have “**includ[ed] an agent for each type of security sensor,**” *i.e.*, a separate conversion module agent for each type of sensor. (Richman, ¶0097 (“Each unique hardware device requires a separate customized conversion module to translate its data into the coded SCA XML format for the purpose of further analysis.”).) The claim term “**agent**” simply refers to software functionality for mediating between the sensor and other components of the system. (Madisetti, ¶80.) Under the proposed combination, therefore, each conversion module is an agent that integrates

sensor data from a security sensor type by converting it into standardized XML format for transmission to the Farino unified access control server. (*Id.*)

Moreover, under the proposed combination, as explained, “**the sensor data from each security sensor is embodied in a native data representation format of each respective manufacturer of the plurality of manufacturers,**” *i.e.*, the unique device-specific language used by the manufacturer of each sensor. (Farino, ¶¶0006 (“In many cases, the signaling protocols [used by ACDs/DCPs] may be specific to a single vendor’s access control products.”), 0009; Richman, ¶0092 (“[S]ecurity devices and sensors transmit data in device language specific for that device or sensor.”).) The sensor data is embodied in “**a native data representation**” because it was encoded using the original format of the data transmitted by the sensor in response to the physical stimulus, or in other words, in the original format that was built-into the sensor. (Madisetti, ¶81.)

An ordinarily skilled artisan would have been motivated to make this combination. (Madisetti, ¶¶83-86.) Farino does not disclose converting native sensor data into an *industry-standard* format; it instead converts into “another access control system vendor’s component formats.” (Farino, ¶0040.) But Richman specifically discloses conversion into standardized XML format – and the many reasons to do so. In a section entitled “XML advantages,” Richman identifies several known benefits of using XML format including (among others) that it “[m]akes it

easier for people to process data using inexpensive software” and “[p]rovides a standard packaging/transport mechanism for any type of information” (Richman, ¶¶0031-0038, 0036, 38.) Richman further explains that “XML is a low-level syntax for representing structured data,” which can be used “to support a wide variety of applications.” (Richman, ¶0043.) “XML as a simple method for data representation and organization will mean that problems of data incompatibility and tedious manual re-keying will become more manageable.” (Richman, ¶0044; *see also id.*, ¶0045.)

The benefits of XML format expressly described in Richman were well-known. (Madisetti, ¶¶83, 85.) An ordinarily skilled artisan would have understood that XML provides a straightforward, standards-based, and easily understandable format for encoding sensor data. This would in turn would have improved interoperability by allowing a larger range of hardware and/or software to receive and process information provided by sensors from disparate manufacturers. These advantages would have been further enhanced by the existence of so many different manufacturers for sensor types, as explained for 11[a]. XML format would thus have been a natural choice for Farino for conversion of the native data provided by security sensors. (Madisetti, ¶85.) A skilled artisan would also have also had a reasonable expectation of success. (Madisetti, ¶86.)

(c) **“defining individual user profiles and their respective access privileges and credentials in the system;” (Claim 11[c])**

Farino discloses several ways in which individual user profiles and respective access privileges and credentials are defined in the system. For example, Farino discloses a table and/or list identifying individual users and their respective access privileges. (Farino, ¶¶0026, 0084, 0097.) This table or list defines access privileges and credentials for network and physical security devices in the system:

A table associated with the unified access control server **200** contains information to support both physical security access control systems and network access control systems. This table includes but not limited to: a full or partial list of authorized users for physical access to a given facility or resource and/or network resource access, user identification numbers, time of day, day of week, network addresses, biometric information, authorization codes and similar information as well as physical access control policy information and network access control policies.

(Farino, ¶0084.) Additionally:

Note that the present invention enables a common list and/or table or synchronized set of lists/tables containing authorized entities, defined by credentials (identity establishment via password, fingerprint, badge, etc.), permitted to access a physical or network facility/resource based upon a specified set of conditions (policies).

(Farino, ¶0097.) A system administrator in Farino can define access privileges and

credentials using “appropriate computer software code, tables and lists and executed at the server **400**.” (Farino, ¶0098.)

Farino thus discloses “**defining individual user profiles and their respective access privileges and credentials in the system**,” with the defined “**individual user profiles**” corresponding to access-related information (in tables or lists) for individual users who can be individually identified by user names or user IDs. (Farino, ¶¶0026 (“This server’s table can hold user names, user IDs, network passwords and rules associated with each user or device that may require access the network.”), 0084; *see also id.*, Fig. 9 (step 388 (“Update user profile”), ¶0093 (“[u]pdate the user’s profile so that their personal preferences follow them from site to site within a facility.”).) Each individual user profile would thus have included its own respective access privileges and credentials. The “**access privileges**” in Farino thus correspond to the facilities or resources the user is permitted to access, while the “**credentials**” correspond to the identity establishment credentials such as the user identity along with the user’s secret password, fingerprint information, etc. required for access to those facilities or resources. (Madisetti, ¶88.)

- (d) **“mapping the sensor data from each security sensor in the native data representation format of each manufacturer of the plurality of manufacturers to a common data representation format, the common data representation format including a data object and processing information for the sensor data;” (Claim 11[d])**

The requirements of claim 11[d] were also discussed for claim 11[b] above. As explained, under the combination of Farino and Richman, the sensor data in the native representation format of each vendor is mapped to XML format (**“a common data representation format”**) by an integration module. (Madisetti, ¶¶79-81.) As explained, Richman discloses a series of conversion modules that take native sensor data from sensors from different manufacturers and convert it to standardized XML format. (Richman, ¶0097.) In one example, a tenant seeks access to a room secured with a keypad and enters a username and password. (Richman, ¶¶0120-0121.) Richman converts the received sensor data into the following standardized XML:

```
<sensor code = "Authorization procedure"
  <event code = "login">
    <attributes>
      <attribute code="Username" value="User1 ">
      <attribute code="Password" value="Guest">
      <attribute code="time" value="3:00 PM">
      <attribute code="room" value="5 ">
    </attributes>
  </events>
</sensor>
```

(Richman, ¶¶0121-0122.) The XML in this particular example identifies the room

number (5) and time, as well as the username and password obtained from the keypad. This XML is then transmitted for processing to the SCA. (Richman, ¶¶0122-0124.)

Richman discloses another example in which a tenant seeks to access a room secured with a fingerprint or eye detection system. (Richman, ¶¶0128-0129.) The received sensor data is converted into the following standardized XML format:

```
<sensor code = "Authorization procedure">
  <event code = "login">
    <attributes>
      <attribute code="Fingerprint Data" value= " 01100101001001010
                                                10010010010010010
                                                01010010010010010
                                                00101001001001011
                                                10101010101010010
                                                010010000101111 ">

      <attribute code="time" value="3:00 PM">
      <attribute code="room" value="5 ">
    </attributes>
  </events>
</sensor>
```

(Richman, ¶¶0128-0129.) Richman discloses several other examples, including a sensor having detected a broken window. (Richman, ¶¶0114-0119.)

The XML format in Richman discloses a “**common data representation format**” because it provides a common (*e.g.* standardized) format for encoding sensor data. (Madisetti, ¶92.) Richman repeatedly describes the XML format in

which sensor data is converted as a “standardized” format. (Richman, *e.g.*, ¶¶0038, 0066, 0097-0099; Madisetti, ¶93.) This is consistent with the ’704 patent; dependent claim 13 addressed below, for example, specifically recites converting the native representation format “into XML (Extensible Markup Language) format,” thus confirming that XML qualifies as a “**common data representation format.**”

That XML format is a “**common data representation format**” is also confirmed by the specific Richman examples above showing common XML elements (*e.g.*, “<sensor code>”, “<event code>”, “<attribute code>”) used across different types of sensors (*e.g.* keypad, motion sensors, etc.). (Richman, ¶¶0110-0115, 0118-0119, 0121-0122.) As further explained for claim 11[f] below, this common XML format is used in the combined system to produce normalized event data. (Madisetti, ¶¶93, 132.)

Richman also discloses that “**the common data representation format includ[es] a data object and processing information for the sensor data.**” This can be seen through the examples in Richman. For example, the “**data object**” can take the form of information in the XML messages identifying the user and/or facility that currently seeks access. The XML examples from Richman, for example, identify the “Username” (¶¶0121-0122) and the “room” (¶¶0121-0122, 0128-0129) the user seeks to access. These pieces of information individually or collectively disclose “**a data object... for the sensor data,**” because they identify the object in

the access control database (*e.g.* room, username, etc.) to which the sensor data applies. (Madisetti, ¶94.) This is consistent with the '704 specification, which provides similar examples of data objects. ('704, 11:58-59 (“...data objects representing personnel and/or assets....”).)

With respect to the “**processing information**” included as part of the common data representation format, Richman discloses this limitation in at least two ways. First, processing information includes the information in the XML that will be used to determine if access is permitted, *e.g.*, the “Password” (¶¶0121-0122), or “Fingerprint Data” (¶¶0128-0129) obtained from the user. This qualifies as “**processing information**” because it represents information the system processes to determine if access is permitted (*e.g.*, by matching password or fingerprint data from the sensor against stored credentials). If access is permitted, the door is opened. (Richman, ¶¶0124-0126; *see also* Farino, ¶¶0018, 0038; Madisetti, ¶95.)

Alternatively, the “**processing information**” in Richman can take the form of the “sensor code” and “event code” information in the XML messages (*i.e.*, “Authorization procedure” and “login,” respectively) (¶¶0121-0122, 0128-0129), which tell the receiving system that the message pertains to an attempt at authorization and login. (Richman, ¶0089 (“The attribute code(s) and value(s) **136**, together with the associated sensor code **132** and event code **134** for a given system event **124**, are detected and **processed** by the checkpoint....”).) This also qualifies

as “**processing information**” because it effectively provides a command or directive to the receiving system instructing it to allow access to a user if the password, fingerprint data, etc., match stored credentials for the user. (Madisetti, ¶96.)

- (e) “**generating unique physical access privileges and credentials to exclusively map a defined user profile to a spatial hierarchy of physical sites along with security devices of the system, wherein the unique physical access credentials maintain a common representation of the user’s identity across the plurality of sites and to associate specific user identities with respective actionable events;**” (Claim 11[e])

As explained for 11[c] above, Farino discloses generating “**access privileges and credentials**” to “**map a defined user profile.**” The generation of user profiles containing access privileges and credentials is evidenced, as explained for 11[c], by access-related information in Farino (tables/lists) recording privilege and credential information for each user, each of whom can be individually identified by a user names or user identifier (ID). (Farino, *e.g.*, ¶¶0026, 0084, 0097; *see also id.*, ¶0098 (“Authorized personnel may create credentials and policies....”).)

Farino thus discloses the ability to generate access privileges and credentials to “**map a defined user profile,**” *i.e.*, by specifying access privileges and credentials for a particular user profile. Petitioner will separately below address the language of 11[e] requiring “**unique**” physical access privileges and “**exclusive**” mapping.

Farino further discloses mapping a user profile “**to a spatial hierarchy of physical sites along with the security devices of the system.**” The ’704 patent

does not expressly define “**spatial hierarchy**” but provides examples of security structures such as doors, buildings, lockers and safes. (’704, 5:5-10 (“... an organization’s spatial hierarchy of its global physical security system deployments (doors, buildings, lockers, safes, and so on).”), 18:26-38 (“...doors, buildings, etc...”).) Farino similarly discloses a security system having a spatial hierarchy of physical sites along with security devices of the system. For example, each floor of a building can have various access control devices (ACDs) such as door locks, readers, keypads, motion sensors, among others, as explained. (Farino, ¶0014.) Farino explains that an access control panel (ACP) “may control all of the door panels in a facility, or a particular floor of a multi-floor building or in a particular area of the facility.” (Farino, ¶0016.)¹ The door control panels (DCPs) in Farino, as explained, in turn control individual ACDs such as card readers, keypads, and so forth. (Farino, ¶¶0014, 0063, 0074; *see also* Richman, ¶0120 (describing example of attempt to “access room 5 of a building”).)

It thus would have been obvious that the user profile (with its associated access privileges and credentials) could have been mapped “**to a spatial hierarchy**

¹ ACP, which stands for Access Control Panel, is described in Farino’s discussion of the prior art as a panel for verifying user credentials for access control devices, such as card or badge readers. (Farino, ¶¶0007-0008.) Farino explains that the functions of the ACP would, in the disclosed system, be performed by the unified access control server. (Farino, ¶¶0061, 0062, 0071.)

of physical sites along with the security devices of the system,” such as particular facilities or buildings, the floors within those buildings, and the rooms within those floors, with security devices deployed at each point in this spatial hierarchy. For example, ACDs (such as card readers, keypads, etc.) could have been distributed at building entrances, at particular floors in those buildings, and at particular rooms on those floors. Because the access-related information in Farino at the unified access control server records user privilege and credential information across the site, it maintains information for these security devices. (Madisetti, ¶98.)

Farino also discloses that the **“physical access credentials maintain a common representation of the user’s identity across the plurality of sites.”** Farino discloses that unified access control server includes an access table, as discussed. (Farino, ¶0084 (“A table associated with the unified access control server **200** contains information to support both physical security access control systems and network access control systems.”).) That table, in turn, includes user identification information such as user identification numbers. (Farino, ¶¶0084 (“This table includes but [is] not limited to: a full or partial list of authorized users for physical access to a given facility or resource and/or network resource access, user identification numbers, time of day, day of week, network addresses...”), 0026 (“This server’s table can hold user names, user IDs, network passwords and rules associated with each user or device that may require access [to] the network.”).) The

“**representation of the user’s identity**” thus takes the form of the user name and/or user ID/identification number stored by the unified access control server in Farino.

This provides a “**common representation**” because the user identify is stored and used by a centralized system for managing user access credentials and access control policies. (Madisetti, ¶¶99-100.) Farino expressly describes this as being a common list and/or table:

Note that the present invention enables a **common** list and/or table or synchronized set of lists/tables containing authorized entities, defined by credentials (identity establishment via password, fingerprint, badge, etc.), permitted to access a physical or network facility/resource based upon a specified set of conditions (policies).

(Farino, ¶0097; *see also id.*, ¶0093 (“The present invention implements a unified policy-based network and physical premises access server that operates off a **common database** of user/personnel attributes.”).) Farino thus discloses a “**common representation of the user’s identity**” because user identity stored by the unified access control server in a list or table that is commonly used throughout the system for access across multiple sites (“**across the plurality of sites**”). As Farino explains, “[t]he unified access control server implements access policy that regulates user access to multiple facilities.” (Farino, ¶0083; *see also id.*, ¶0062 (“Server **200** also enforces all access policies.”).)

It is unclear if the requirement of a “**common representation**” further

requires an industry-standard format for storing access privileges and credentials such as XML, as opposed to merely a representation that is common to the system. But even if the claim were interpreted to impose such a further requirement, it would have been obvious based on Richman to store this information in XML format. Richman explains that XML “[p]rovides a standard packaging/transport mechanism for any type of information.” (Richman, ¶0038.) Richman provides an example in which customer information is stored in XML format, which is analogous to the storage of individual user profile information in Farino. (Richman, ¶0040.) It would have been obvious to store the common list and/or table of Farino in standard XML format, and an ordinarily skilled artisan would have been motivated to do so to achieve the known benefits of XML described for 11[b] above, including standardization and improved interoperability. (Madisetti, ¶¶101, 84-85.)

Finally, the physical access privileges and credentials in Farino “**associate specific user identities with respective actionable events.**” As noted, the common list or table in Farino identifies not only the specific user identities, but also the respective facilities or resources the user is permitted to access if the user seeks access (“**actionable events**”). (Farino, ¶¶0084 (“The table includes but [is] not limited to: a full or partial list of authorized users for physical access to a given facility or resource and/or network resource access, user identification numbers....”), 0097 (“.... a common list and/or table or synchronized set of

lists/tables containing authorized entities, defined by credentials (identity establishment via password, fingerprint, badge, etc.), permitted to access a physical or network facility/resource based upon a specified set of conditions (policies).”).)

The “**actionable events**” thus correspond, using the ACD example, to requests to access particular resources or facilities. (Madisetti, ¶102.)

“*Unique*” and “*Exclusive*”: Petitioner notes that 11[e] recites the step of “generating unique physical access privileges and credentials to exclusively map a defined user profile,” and it is possible that “**unique**” and “**exclusively**” may be interpreted to require that a defined user profile have access privileges and credentials that are unique and exclusive to the underlying user, *e.g.*, which are not shared by any other user or user profile in the system. Such a requirement would still have been obvious over the prior art.

With respect to “**credentials**,” it would have been obvious that a defined user profile could have unique and exclusive credentials. As explained above for 11[c], the “**credentials**” in Farino correspond to identity establishment credentials such as the user identity along with its respective password, fingerprints, etc., that required to access a facility or resource. It would have been obvious that such credentials would have been unique and exclusive to a particular user. For example, a user identity with associated fingerprint information would obviously have been unique and exclusive to a particular user. (Madisetti, ¶104.)

With respect to “**physical access privileges**,” it would also have been obvious that a defined user profile could have unique and exclusive physical access privileges. It would have been obvious that a particular user could have been the only user who has been granted access to a particular area or facility. This concept would have been well-known to ordinary skilled artisans and even laypersons, with the concept long existing in popular culture (such as the secret “Fortress of Solitude” accessible only by Superman). In the context of a building or facility, it would have been obvious that a corporate executive, owner, or other individual could have had unique and exclusive access to particular areas or rooms. (Madisetti, ¶105; *see also* EX1008, ¶120 (“For example, a corporate vice president can have her workspace allocated specifically and only to her, thereby allowing her to reserve that workplace as desired.”).) Alternatively, it would have been obvious that a particular user who is trusted or has a special role (such as an administrator or owner) could have had broad privileges for access to all areas within all facilities – a level of access shared by no other user. (Madisetti, ¶105.) Under either theory, it would have been obvious that a defined user profile would have unique and exclusive “**physical access privileges**,” as an ordinarily skilled artisan would have appreciated many scenarios in which it would have been desirable for a defined user profile to have a set of access privileges not shared by any other user of the system. An ordinarily skilled artisan would have been motivated to provide such a defined user profile, as

explained above, to achieve the benefit of giving a particular user a unique and exclusive level of physical access. (Madisetti, ¶105.)

- (f) **“defining physical security policies of the site in the context of user profiles at all sites through actionable representations of physical, network and information technology resources of the site, wherein the security policies define standardized rule definitions through visual rules depicted by live objects that contain attributes to define their spatial relationship to the actionable representations, and that are applied to the actionable events normalized to the common data representation format to produce normalized event data; and” (Claim 11[f])**

In light of the length of this limitation, Petitioner has broken it into parts.

- (1) **“defining physical security policies of the site in the context of user profiles at all sites**

Farino also discloses **“defining physical security policies of the site in the context of user profiles at all sites.”** There appears to be no antecedent basis for **“the site,”** and as such, Petitioner has interpreted 11[f] to require defining physical security policies of one or more sites within the system.

Generally speaking, a policy refers to a rule or set of rules governing access to the resources of an organization, such as its physical premises or network devices and information. (Madisetti, ¶108.) Here, the **“physical security policies of the site”** was largely covered in the discussion of 11[e] above, in which Petitioner explained that the unified access control server maintains a list or table identifying access privileges and credentials. That table or list also includes **physical security**

policies. (Farino, ¶0026, 0084 (“This table includes but [is] not limited to: ... physical access control policy information and network access control policies.”), 0097 (“Note that the present invention enables a common list and/or table or synchronized set of lists/tables containing authorized entities, defined by credentials... permitted to access a physical or network facility/resource based upon a specified set of conditions (policies).”).) The unified access control server, in one embodiment, includes a policy engine that “includes the capability for implementing, enforcing, and executing physical and/or network access control policies (policy enforcer) which generates and transmits policy-based responses based on given event(s) to other devices for a corresponding, pre-defined action.” (Farino, ¶0098; *see also id.*, ¶0018 (“If the policy indicates that the door should be opened, an access control response message is transmitted... to instruct it to activate (open)... that specific door.”).) Farino thus discloses “**physical security policies**” in the form of policies specifying rules required to gain physical access to a site or facility. (Madisetti, ¶108.)

Farino further discloses physical security policies “**of the site in the context of user profiles at all sites.**” Farino discloses that a physical security policy can pertain to a particular site or facility (“**of the site**”). (Farino, ¶0090 (“The unified access control server **200** determines the access policy for the particular combination of the credential and physical location...”).) And these physical security policies,

even for an individual site, are also defined “**in the context of user profiles at all sites.**” This is because, as explained fully for 11[d] above, Farino discloses a centralized and unified access control server that stores access privileges, credentials, and security policies for all users and all sites. (Farino, ¶0093 (“The present invention implements a unified policy-based network and physical premises access server that operates off a common database of user/personnel attributes.”), 0061, 0094 (“A unified policy engine **410**, which is a part of unified access control server **400**, acts as the point of unification for lists and policies that tie physical access policies and events to network policies and events.”), 0083 (“The unified access control server implements access policy that regulates user access to multiple facilities.”), 0075 (“[T]he unified system... may include multiple facilities...”)).

The physical security policies defined with respect to a particular site, therefore, are also defined “**in the context of user profiles at all sites,**” because all policies can be stored (for example, in a single table) within the unified access control server along with all user profiles, access privileges, and credentials. (Farino, *e.g.*, ¶¶0086 (“In one embodiment, the physical access control and network access control tables are consolidated into a single table that credential verification and policy enforcement engine(s) access.”), 0084; Madisetti, ¶109.)

Finally, with respect to “**defining physical security policies,**” Farino explains that “[a]uthorized personnel can create credentials and policies,” at a secure

management station executed at the unified access control server. (Farino, ¶0098.)

- (2) “...through actionable representations of physical, network and information technology resources of the site, wherein the security policies define standardized rule definitions through visual rules depicted by live objects that contain attributes to define their spatial relationship to the actionable representations, and”

Neither Farino nor Richman disclose details about the user interface that allows administrators to define security policies. **Ground 1** thus cites Farino and Richman in combination with **Wiegel** (EX1005), for these other aspects of claim 11[f]. Wiegel discloses a graphical user interface for managing security policies in a network-based system. (Wiegel, *e.g.*, 1:12-16.) Figure 3 shows the user interface:

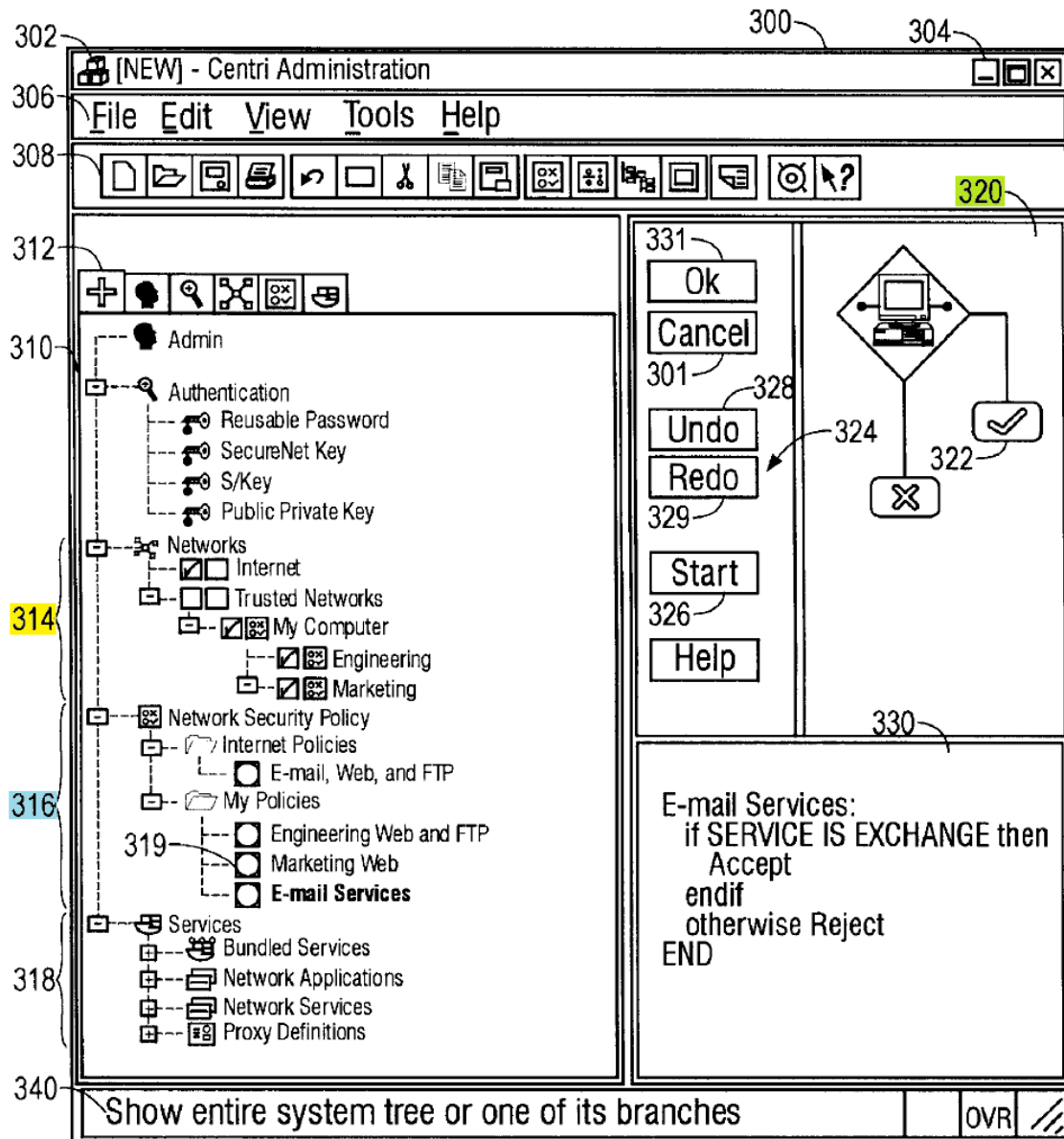


FIG. 3

(Wiegel, Fig. 3 (highlighting added).) Figure 3 shows a user interface window 300 which includes a number of different components. Network tree 314 shows “the hierarchical relationship of each network that is physically available in the user’s environment.” (Wiegel, 15:29-32.) Policy tree 316 “represents previously established security policies that are available to be applied to nodes of a network.”

(Wiegel, 15:43-46.) “To establish a security policy applicable to a network or one of its nodes,” Wiegel explains, “the user can drag network security policies and drop them onto each icon in the network tree **314**.” (Wiegel, 15:38-42; *see also id.*, 27:26-40.) Figure 9 shows that, after the administrator drags a security policy **319** into an icon in network tree **314**, the network tree icon can be shown alongside a policy icon indicating that the network resource is governed by the policy:

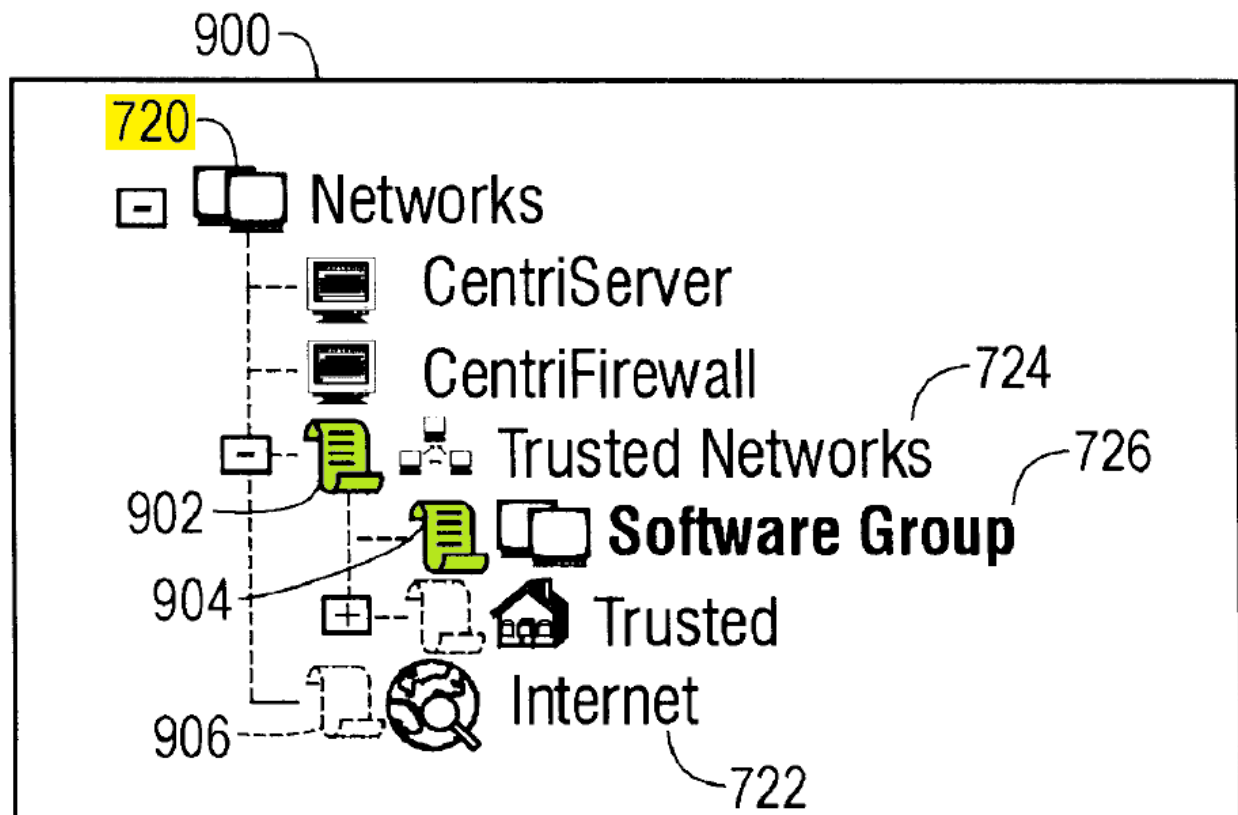


FIG. 9

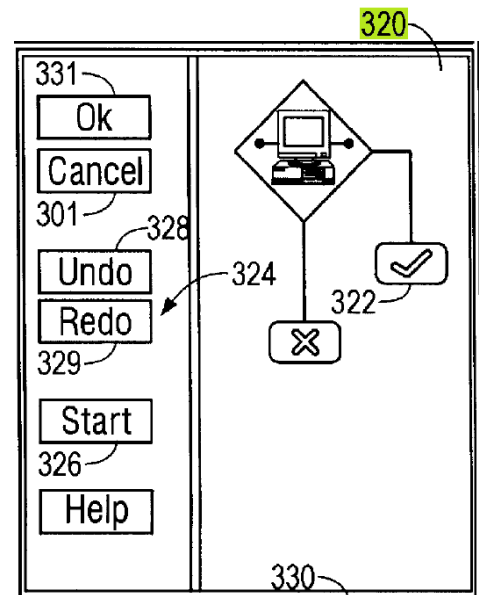
(Wiegel, Fig. 9 (highlighting added), 27:34-40.) Figure 9 shows networks tree **720** including policy icons **902** and **904** (shown in **green** highlighting), which “indicate that security policies have been applied to the Trusted Networks object **724** and

Software Group object **726**, respectively, using the [drag-and-drop] process described above.” (Wiegel, 27:43-46.)

Moving back to Figure 3, the top right of window **300** shows the policy builder pane **320** (also shown excerpted at right), which is used to actually construct a

security policy. (Wiegel, 16:2-8.) “The user constructs a security policy by arranging one or more icons **322** into a representation of the policy in the policy builder pane **320**.” (Wiegel, 16:5-8.)

“As the security policy is constructed, the administration component **206** generates a source script that defines the policy in a scripting



language, and displays the script in the script pane **330**.” (Wiegel, 16:9-12.) Wiegel thus allows the user to construct a security policy without having to manually write the corresponding scripting language code.

The policy builder in Wiegel supports two different types of symbols that can be displayed in builder pane **320**: (1) **condition** symbols and (2) **action** symbols, examples of which are shown in Figure 4A:

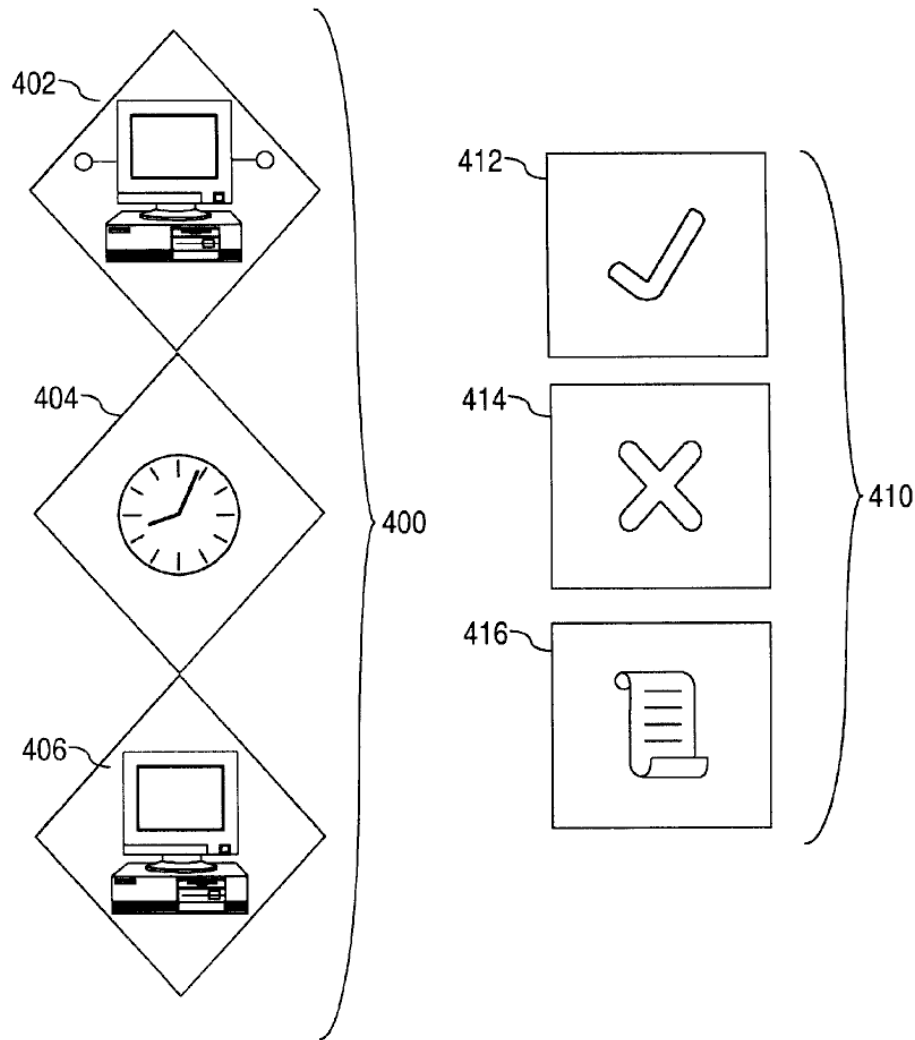


FIG. 4A

(Wiegel, Fig. 4A.) Wiegel explains that “[t]he condition symbols **400**,” shown on the left of Figure 4A above, “are represented using diamond-shaped icons, and comprise a service condition symbol **402**, a time range symbol **404**, and a destination symbol **406**.” (Wiegel, 16:40-43; *see also id.*, 16:43-56 (describing symbols).) Other condition symbols may be provided. (Wiegel, 16:55-56.)

Figure 4A, on the right, also depicts three action symbols **401**. (Wiegel, 16:37-39, 16:57-65.) Wiegel explains that an “action” symbol refers to “a component of a

security policy that resolves a condition.” (Wiegel, 16:66-17:2.) Figure 4A shows three action symbols, *i.e.*, Accept symbol **412**, a Reject symbol **414**, or Use Next Policy symbol **416**. (Wiegel, 16:17-65.) As explained below, these symbols are used to create the claimed visual rules.

Having outlined the disclosures of Wiegel, Petitioner will first explain the manner of combining the Wiegel to Farino and Richman, and then, describe the motivations an ordinarily skilled artisan would have had to do so.

Manner and Rationale to Combine (Farino and Richman with Wiegel): It would have been obvious to combine Farino and Richman with Wiegel, predictably resulting in the Farino-Richman system in which the physical security policies of the site were defined “**through actionable representations of physical, network and information technology resources of the site.**” Under the proposed combination, “**actionable representations**” take the form of icons displayed in the user interface for various resources of the site (in network tree **314** of Figure 3 and network tree **720** of Figure 9 of Wiegel). Referring again to Figure 9 of Wiegel:

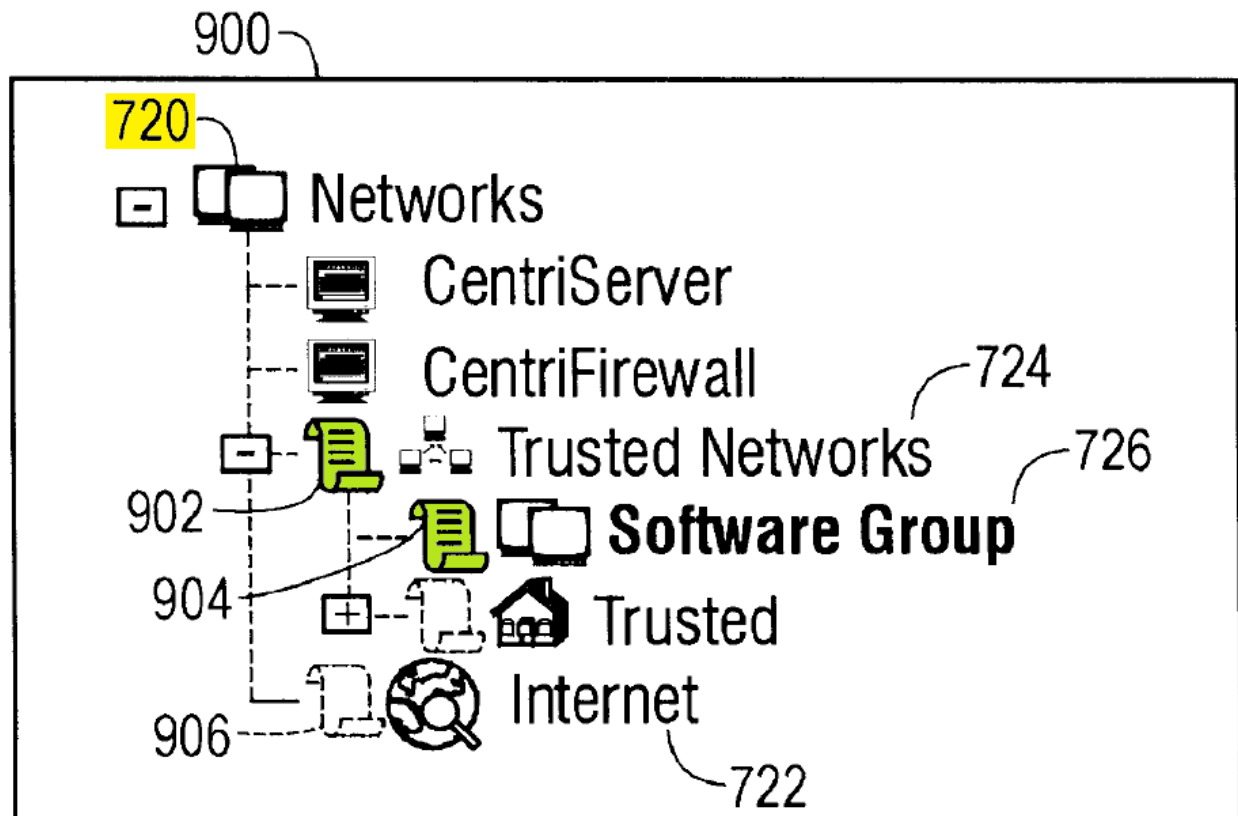


FIG. 9

(Wiegel, Fig. 9 (highlighting added).) The icons in the network tree above (in this

case,  and ) represent at least “**network and information technology**

resources of the site,” as claimed, because they represent networks (Trusted

Networks 724) and information technology (IT) resources (Software Group 726).

With respect to “**physical... resources of the site,”** under the proposed combination

with Farino and Richman, the user interface would have been further adapted to

show icons and corresponding textual labels for physical security resources, such as

access control devices (ACDs) and the other security devices in Farino discussed

above. (Madisetti, ¶118.) Under the proposed combination, the icons representing

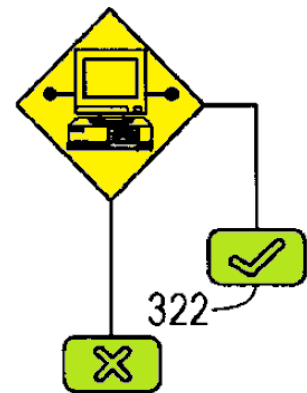
network, IT, and physical resources qualify as “**actionable representations**” because user actions can be taken with respect to them; for example, the administrator can drag-and-drop a security policy onto an icon to associate it with the policy. (Wiegel, *e.g.*, 14:44-47, 15:38-41, 15:60-63, 27:36-40.)

Additionally, under the Farino-Richman-Wiegel combination, “**the security policies define standardized rule definitions through visual rules depicted by live objects that contain attributes to define their spatial relationship to the actionable representations.**” The security policies define “**standardized rule definitions,**” for at least two reasons. First, as explained above, they are written in a common scripting language. (Wiegel, 14:12-19.) Wiegel explains that scripts can be prepared “in a language called PolicyScript™, a policy language that is based on English.” (Wiegel, 14:15-17.) Although Wiegel uses PolicyScript™ as an example, an ordinarily skilled artisan would have appreciated that any other well-known standardized scripting language (*e.g.* PERL, JavaScript, etc.) could have been used. (Madisetti, ¶119.) In any case, because all of the security policies (and their underlying rules) under the proposed combination would have been represented using a common scripting language, the security policies define and adhere to “**standardized rule definitions.**” (*Id.*) The rule definitions are also standardized because, as explained, they are all created using the user interface of Wiegel that provides a consistent format for specifying conditions and actions using standard

visual symbols. (Wiegel, 18:62-65 (“The structure of the symbols displayed in the policy builder pane 320 reflects certain rules followed by the administration component 206 to enforce policy consistency and reliability.”).) Because the policies are created using consistent user interface symbols and a common scripting language, they define standardized rule definitions as claimed. (Madisetti, ¶119.)

The security policies also define these standardized rule definitions “**through visual rules depicted by live objects,**” for example, through the process above of using condition and action symbols to define rules.

In the pending litigation, Petitioner has asserted that a “**visual rule**” is “*a rule that is visually defined and depicted in a display area by placing icons and connectors in appropriate relationships to one another,*” and that “**live objects**” are “*stencil shapes that are used to represent normalized systems, data, and processes.*” Patent Owner disagrees and believes these terms should be given their plain meaning. The proposed combination discloses these limitations under either view, as the icons in Wiegel and the way in which are arranged disclose both the “**visual rules**” and “**live objects**” limitations. Using the example from the policy builder pane of Figure 3 (excerpted at right), the policy in this case includes visually defined rules depicted using **condition** and **action** symbols (icons) placed and connected to each other in appropriate relationships. These



icons also qualify as stencil shapes, as they have standardized and consistent shapes providing building blocks for representing systems, data, and processes such as services. ('704, 11:13-14 (“Such building blocks are also referred to as stencil shapes or live objects.”).) The highly-simplified rule from Wiegel from Figure 3 simply checks to see if the service is “exchange.” (Wiegel, Fig. 3 (330); *see also id.* Fig. 5 (showing more complex example).) The underlying systems, data, and processes upon which the rules act are normalized because, as explained, the rules are created using a graphical user interface with standardized symbols and expressed via policies in a common/standard scripting language. (Madisetti, ¶¶121, 119.)

These visual rules and live objects also “**contain attributes to define their spatial relationship to the actionable representations.**” This is because the visual rules that define the security policies under the proposed combination are associated with the icons (“**actionable representations**”) for the physical, network and information technology resources of the site. Referring again to Figure 9:

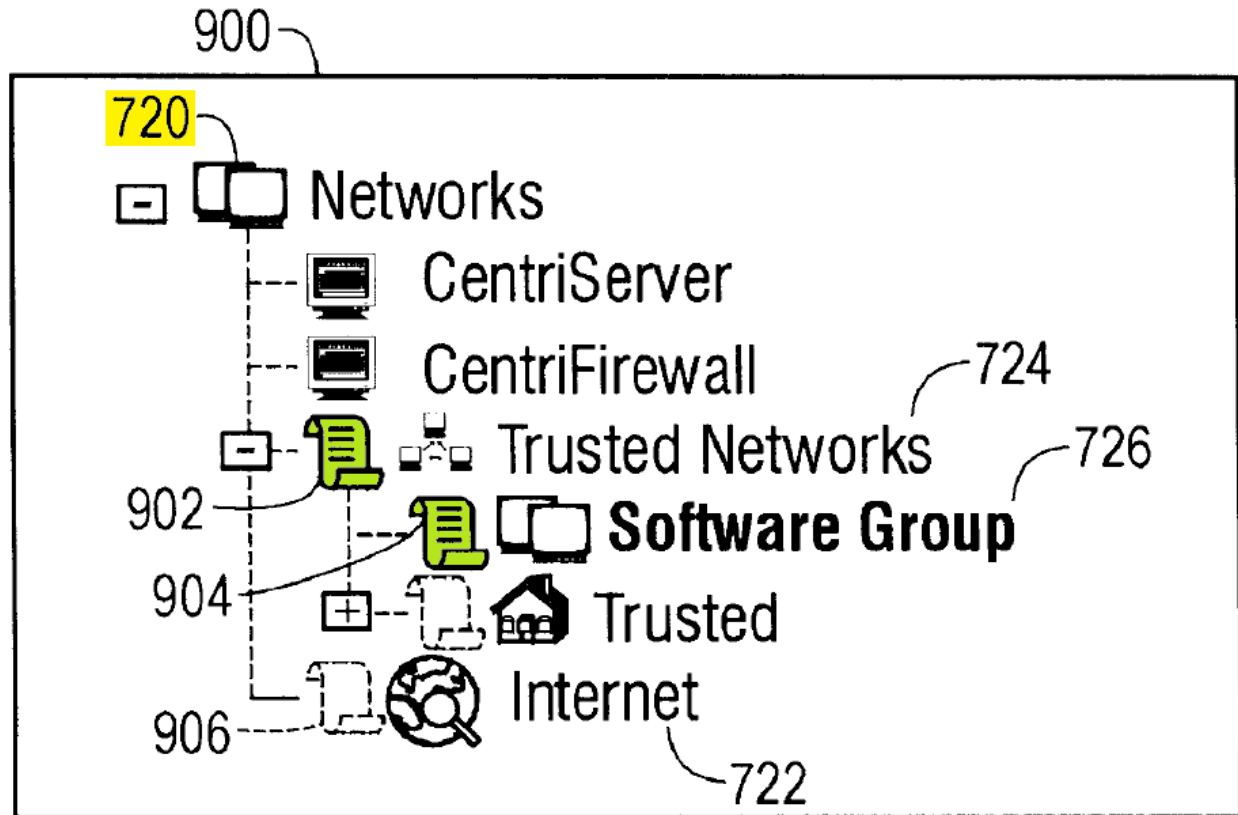


FIG. 9

(Wiegel, Fig. 9 (highlighting added).) In this example, as explained, security policies in network tree 720 are shown in green, and they embody visual rules created using the Wiegel user interface as explained. Figure 9 plainly shows they are spatially associated with their neighboring icons (“**actionable representation**”) representing the physical, network and information technology resources to which the policies have been applied (by using drag-and-drop as described above). As also noted above, the user interface screen in Wiegel (Fig. 3) also includes builder pane 320 that displays the particular security policy 319 that was applied (Wiegel, Fig. 3, 27:31-46), and as such, the visual rules and live objects would also have been visible

to user on the display screen. (Madisetti, ¶122.)

Wiegel does not disclose how the system internally organizes security policies and their associations with resources as displayed in network tree 720, but it would have been obvious to any ordinarily skilled artisan that the system would have included data (attributes) such as metadata to record that security policies have been applied to such resources, to ensure that the system has a way of knowing which security policies have been applied to which resources. (Madisetti, ¶123.)

Wiegel therefore discloses and renders obvious that the visual rules and live objects “**contain attributes to define their spatial relationship to the actionable representations,**” as claimed. Now having explained how Farino and Richman would be combined with Wiegel, Petitioner will explain the motivation to do so.

Motivation to Combine (Farino and Richman with Wiegel): An ordinarily skilled artisan would have been amply motivated to combine in the manner described above. As a threshold matter, like Farino and Richman discussed above, Wiegel is also an analogous reference. (Madisetti, ¶¶126, 82.)

For the motivation to combine, as explained previously, Farino contemplates that the unified access control server will have software for allowing an administrator to define physical security policies, but Farino does not disclose further details. (Farino, ¶0098.) An ordinarily skilled artisan would have been motivated to consult Wiegel because it provides an easy-to-use user interface that

allows the administrator to create security policies based on intuitive graphical decision-trees, rather than having to manually write complex code in a scripting language. (Wiegel, 2:32-35 (“Presently, human network administrators are not generally trained in computer programming.”), 31:5-7 (“By using decision trees to construct security policies, administrators can define security policies they understand.”).) Wiegel provides a further express motivation to combine by further explaining that “[m]anagement of a large, complicated network made up of diverse devices is made easier by mechanisms that allow information about all policies for devices in the network to be viewed at a graphical workstation.” (Wiegel, 3:52-55.) An ordinarily skilled artisan would thus have been motivated to combine the Farino-Richman physical security with the teachings of Wiegel to provide a straightforward user interface to easily define security policies for the site. (Madisetti, ¶127.)

Wiegel concerns security policies for computing resources such as networks and routers, but this provides no obstacle to the proposed combination. An ordinarily skilled artisan would have recognized that the techniques for defining security policies in Wiegel could have been applied to a wider array of network-connected devices, including the security sensors and access control devices (ACDs) of Farino. An ordinarily skilled artisan would thus have been motivated adapt the Farino-Richman system, based on Wiegel, to provide a graphical user interface for creating security policies for both physical and network security devices of the site.

(Madisetti, ¶128.) An ordinarily skilled artisan would have had every expectation of success. (Madisetti, ¶129.)

(3) **“...that are applied to the actionable events normalized to the common data representation format to produce normalized event data; and”
(Claim 11[f])**

The final clause of 11[f] adds little that was not already covered above. Under the proposed combination, the **“standardized rule definitions,”** embodied in security policies created using the Farino-Richman-Wiegel system as discussed, **“are applied to the actionable events normalized to the common data representation format,”** as claimed. This occurs, as explained below, when the unified access control server executes the rules embodied in the security scripts.

The **“actionable events,”** as described for 11[e], correspond for example to requests to access particular resources or facilities. As explained for 11[d] above, those actionable events were **“normalized to the common data representation format to produce normalized event data,”** *i.e.*, XML format resulting from conversion of event data from its native (vendor-specific) format to XML format under the conversion techniques of Richman. (Madisetti, ¶¶79-80, 89-92, 132.) Under the proposed combination, therefore, the security policy would have been **applied** to the actionable events which are represented as normalized event data:

By way of example, when unified access control server 200 receives an access request message from a computer 151 or a DCP 119 connected

to access control gateway **225** via frame-/packet-based network **228**, the unified access control server **200** responds to the request message by verifying the validity of the credential, the location of the requesting device or requestor, and executing a specific policy based on the information provided. The result of executing this access control policy is a corresponding access control response message that is sent back to the computer **151** or the DCP **119** attached to access control gateway **225** via network the frame-/packet-based network.

(Farino, ¶0067.) This response can be implemented, for example, “by either setting off the alarm or opening the door.” (Farino, ¶0064; *see also* Richman, ¶¶0124 (describing return XML message granting access), 0126 (“The checkpoint decrypts and processes the received XML code and opens the door.”).)

- (g) **“receiving the normalized event data and applying relevant transformation and routing rules comprising condition-action sequences in order to maintain user profiles and physical security states across the plurality of sites and to resolve the actionable events through the associated specific user identities.” (Claim 11[g])**

Limitation 11[g] is largely covered by the discussion of claim 11[f] just provided. As noted, the normalized XML event data under the proposed combination is received and processed by applying a security policy.

That security policy in turn comprises **“relevant transformation and routing rules comprising condition-action sequences.”** Petitioner has asserted in the pending litigation that a **“routing rule”** is *“a rule that controls the flow of data*

between different connected systems,” and a “**transformation rule**” is “*a rule that defines and maintains the transformational mapping of data and events between different connected systems.*” Patent Owner does not agree with these constructions and believes the phrase “**transformation and routing rules**” should not be separated and should be given its plain and ordinary meaning. The prior art renders obvious the claimed rules under either view.

The security policies clearly provide **routing** rules because they control the flow of data between different connected systems. Using the example from ¶0067 of Farino quoted above, “when unified access control server 200 receives an access request message... the unified access control server 200 responds to the request message by... executing a specific policy based on the information provided,” and as a result, “a corresponding access control response message that is sent back... via network the frame-/packet-based network.” (Farino, ¶0067; *see also* Farino, ¶0124 (describing return XML message granting access).) The security policies thus provide **routing** rules because they control data flow between unified access control server and other network-connected systems, *e.g.*, security devices requesting access from the unified access control server. They also provide **transformation** rules because they process and transform/map incoming event data (*e.g.* an “access request message” in Farino as discussed above) to appropriate outgoing response data (*e.g.* “a corresponding access control response message” in Farino) sent over

the network. (Farino, ¶0067; Richman, ¶¶120-0126.) As explained, the unified access control server processes and transforms access request messages into appropriate access control return messages in accordance with the security policy. (Madisetti, ¶135.)

A skilled artisan would have appreciated that security policies under the proposed combination comprise “**condition-action sequences**,” *i.e.*, logic specifying actions be taken if specified conditions are satisfied. (Madisetti, ¶136.) The conditions and actions correspond, respectively to the policy conditions required for access and the action of allowing access if they are met. (Farino, ¶0097 (“Note that the present invention enables a common list and/or table... containing authorized entities, defined by credentials... permitted to access a physical or network facility/resource based upon a specified set of conditions (policies).”).) This is the basic IF-THEN structure that underlies all types of computer programs. (Madisetti, ¶136.) In Wiegel, the security policy is explicitly defined via sequence of conditions and actions. (Wiegel, *e.g.*, 4:63-65 (“The representation is established in the form of a decision tree that is constructed by assembling graphical symbols representing policy actions and policy conditions.”), 18:1-5.)

Finally, under the proposed combination, the rules are applied in order “**to maintain user profiles and physical security states across the plurality of sites to resolve the actionable events through the associated specific user identities.**”

This is because the security policies are used, in conjunction with the user profiles and credentials, to determine how to respond to (“**resolve**”) requests for access (“**actionable events**”), such as whether or not to permit access. (Farino, ¶¶0097, 0084, 0026, 0092, 0068.) The system bases its response on “**the associated specific user identities**,” *i.e.*, the identity of users requesting access and their associated credentials as noted above. (Farino, ¶¶0067, 0097.) The response can result, for example, in either “setting off the alarm or opening the door.” (Farino, ¶0064; *see also* Richman, ¶¶0124 (describing return XML message granting access), 0126 (“The checkpoint decrypts and processes the received XML code and opens the door.”).) This “**maintain[s] user profiles and physical security states**” by ensuring that requests for access are allowed only if permitted based on the user profile, thus safeguarding the physical security of the site and ensuring that access privileges in user profiles are enforced and maintained. This occurs “**across the plurality of sites**” because, as explained in detail above, the unified access control server in Farino manages user privileges, credentials, and policies across all sites. (Madisetti, ¶¶99-100, 108-109; Farino, *e.g.*, ¶¶0069, 0083-0084, 0086, 0093.)

2. **Claim 12: The method of claim 11 wherein the plurality of sensor types comprise physical access, intrusion, and environmental conditions, and wherein the action events comprise at least one of generating an audible alarm, notifying authority personnel, initiating lockdown mode of the site, initiating computer backup and shutdown procedures, and facilitating exit from the site.”**

As explained fully for claim 11[a], the plurality of sensor types comprises at least card readers (“**physical access**”), motion sensors (“**intrusion**”), and fire or smoke detectors (“**environmental conditions.**”). (Farino, *e.g.*, ¶¶0063 (“card readers” and “motion sensors”), 0079 (“fire and smoke sensors”); *see also* Richman, ¶0090 (“... the integrated security system can communicate with any off the shelf security device, such as motion sensors, etc., as well as with any facilities subsystem monitoring devices, such as climate control or fire and smoke detection devices.”).)

With respect to “**action events**,” claim 12 requires the presence of only one of the listed examples but Farino discloses at least two. First, Farino discloses “**action events**” that comprise at least “**facilitating exit from the site.**” For example, “in the event of an emergency such as a fire or explosion, corporate security and rescue personnel need to know who is in the facility and where they are located so that they can be quickly and safely evacuated.” (Farino, ¶0047.) Second, Farino discloses “**action events**” comprising “**generating an audible alarm.**” (Farino, ¶0064 (“...for example, setting off the alarm....”).)

3. **Claim 13: “The method of claim 11 further comprising converting the native data representation format for each security sensor into XML (Extensible Markup Language) format, and wherein the native data representation format encompasses native sensor data including text data and binary data represented in a proprietary format, and wherein the native sensor data includes at least one of a command or sensor state information.”**

As fully explained for claim 11[d], under the proposed combination, the **“native data representation format for each security sensor”** is converted into XML format as disclosed in Richman. The remaining aspects of claim 13 address characteristics of the **“native data representation format.”**

Farino explains that the native format for each sensor type can be, for example, a proprietary format used to communicate sensor data from access control devices (ACDs), as shown in Figure 3:

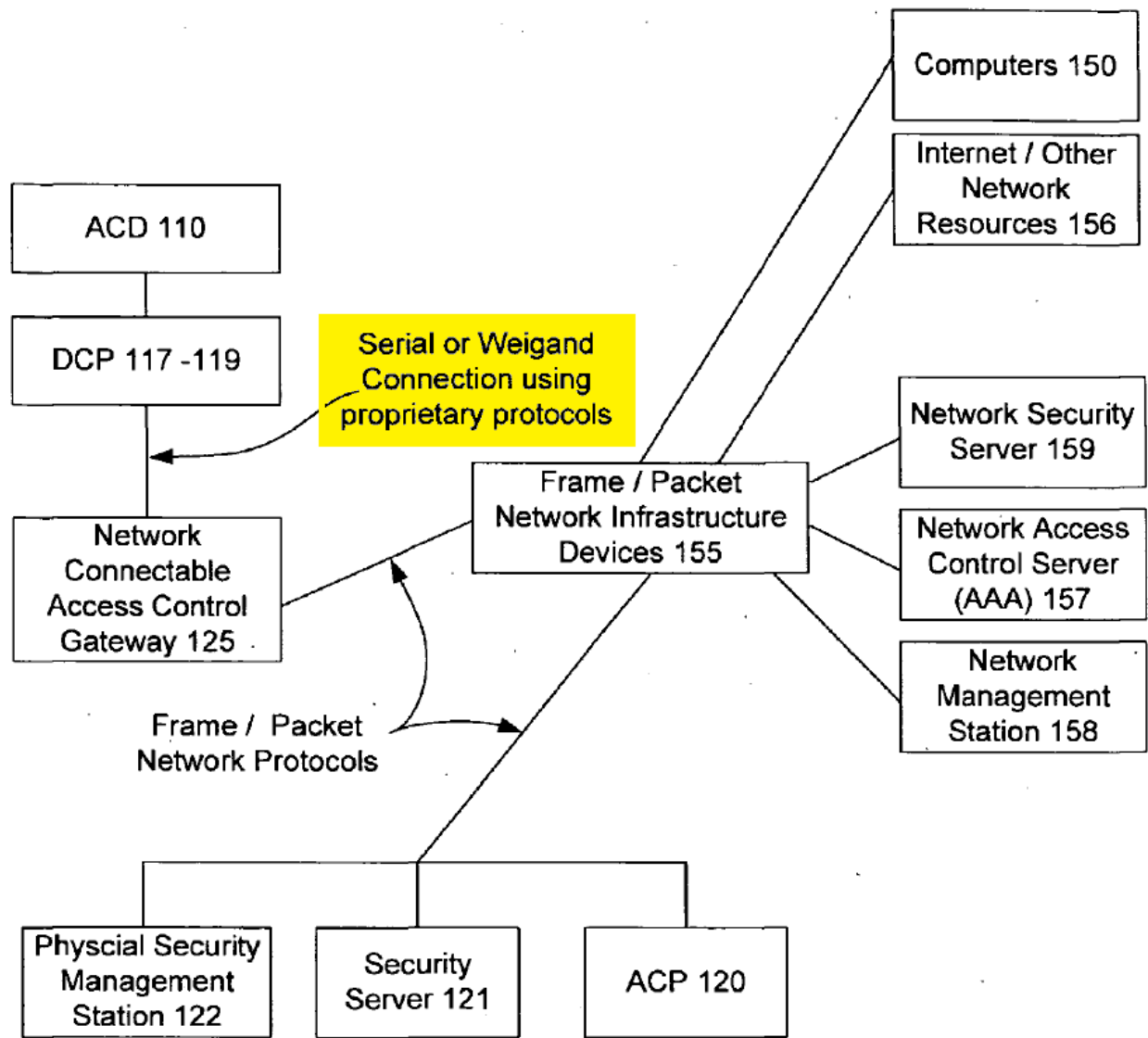


FIGURE 3
(PRIOR ART)

(Farino, Fig. 3 (highlighting added).) As shown, access control devices (ACDs) 110 and door control panels (DCP) 117-119 communicate using a “Serial or Weigand [sic; Wiegand] connection using proprietary protocols.” (*Id.*; see also *id.*, ¶0006 (“It is common for ACDs... for a given entrance to be connected by a serial Wiegand

connection....”).) The native format is thus “**represented in a proprietary format,**” as claimed. This is why Figure 3, as shown above and discussed extensively for claim 11, includes access control gateway **125** that can “translate various access control system device vendor’s signaling formats and protocols to another access control system vendor’s component formats.” (Farino, ¶0040.)² As explained further below, the native sensor data includes **text data** and **binary data**.

The proprietary “Wiegand” format referenced in Farino refers to a well-known technique for sending information from devices such as access control card readers, which includes a number of bits including “data” and “parity” bits. (Madisetti, ¶141 (citing EX1009, ¶0003).) It would have been obvious that the native format in Farino would have “**encompasse[d] native sensor data including text data and binary data,**” as claimed. With respect to “**binary data,**” this would have included bits transmitted in Wiegand format, such as at least parity bits used for error correction. (Madisetti, ¶141.) Binary data is also fundamental to the representation of data in all computer-based systems. (*Id.*)

It would also have been obvious that the native sensor data for each sensor type would have also included at least “**text data.**” This is because, under the

² Although Figure 3 reproduced in the text includes a “prior art” label, as explained for claim 11[b], access control gateway **125** in Figure 3 is a component of the invention of Farino and used to translate proprietary (vendor-specific) sensor data for the unified access control server. (Farino, ¶0063, Fig. 7A (**125**).)

proposed combination, the original (native) data provided by the sensor may have included textual data. (Madisetti, ¶142.) For example, Farino and Richman disclose examples in which a user provides a textual user name and password at a keypad sensor device. (Farino, ¶0005; Richman, ¶0120.) Because the original sensor data collected at the sensor can expressly include text information obtained from the user, the native sensor data would have incorporated that text data in order to allow the unified access control server of Farino to process it (*e.g.*, by determining if the user-entered text data matches stored credentials for that user). (Farino, ¶0038 (“If the user name and password match an entry in a network access list/table, the user is granted network access privileges.”); Madisetti, ¶142.)

Finally, it would have been obvious that “**the native sensor data includes at least one of a command or sensor state information.**” For example, the native sensor data would have included at least “**a command,**” such as a request for access to a resource. (Farino, ¶0067 (“By way of example, when unified access control server **200** receives an access request message...”); Richman, *e.g.*, ¶¶0121, 0128 (“<event code = ‘Login’>”); Madisetti, ¶96.) The native sensor data would have also included at least “**sensor state information,**” such as the information obtained from the sensor through the user interaction, as explained above, such as the credential information provided at the sensor by the user. (Madisetti, ¶¶143, 94.)

4. **Claim 14:** “The method of claim 11 further comprising applying, through a rules engine, defined rules on the data and commands from each security sensor, wherein each rule has an associated action and one rule is applied at a time to the normalized event data in a sequential order dictated by one of a defined rule execution order or a top to bottom order based on event time.”

Claim 14 does not add much to what was already discussed for claim 11[g], in which Petitioner explained that the unified access control server in the proposed combination applies the rules in the security policy to the normalized event data from the security sensors. (Madisetti, ¶¶144, 132.) The “**defined rules**” and “**rules engine**” of claim 14 therefore correspond, respectively, to the conditional rules in the security policy and the software functionality at the unified access control server to execute those defined rules. (Farino, ¶¶0062 (“Server **200** also enforces all access policies.”), 0092 (“Server **200** executes the associated access control policy for the computer based on verification of the credentials.”).)

Each rule is also applied on “**data and commands from each security sensor**.” As explained for claim 11[d], the normalized event data from the security sensor includes at least the identification of users and their associated credentials (“**data**”) along with their requests for access (“**commands**”). (Farino, *e.g.*, ¶0067 (“By way of example, when unified access control server **200** receives an access request message... the unified access control server **200** responds to the request message by verifying the validity of the credential, the location of the requesting

device or requestor, and executing a specific policy based on the information provided.”).) And as explained above, each rule has “**an associated action**” to take as a result of applying the rule, *e.g.*, allow access if the credentials provided through the sensor are valid. (Farino, ¶¶0098; *see also id.*, ¶¶0018 (“If the policy indicates that the door should be opened, an access control response message is transmitted... to instruct it to activate (open)... that specific door.”), 0038 (“If the user name and password match an entry in a network access list/table, the user is granted network access privileges.”); *see also* Richman, ¶¶0124-0126; Madisetti, ¶145.)

Under the proposed combination, “**one rule is applied at a time to the normalized event data in a sequential order dictated by one of a defined rule execution order or a top to bottom order based on event time.**” This limitation merely states a basic property of the sequential execution order of any computer program. (Madisetti, ¶146.) Computers generally execute programs in sequential order, *i.e.*, each instruction is carried out in the order in which it appears in the program. This characteristic also allows programs to include sequences of nested conditional rules in order to create more complex logic. (*Id.*) Wiegel, for example, discloses the ability to link multiple rules in a security policy through multiple nested multiple IF-THEN statements for execution in a particular sequence. (Wiegel, 18:1-24.) The ability to sequence such condition-action sequences in this manner was elementary knowledge to ordinarily skilled artisans. (Madisetti, ¶146.) Under the

proposed combination, therefore, **“one rule is applied to the normalized event data in sequential order,”** *e.g.*, one condition-action sequence in the security policy is applied normalized event data in a sequential order, dictated by at least **“a defined rule execution order,”** *i.e.*, the order specified by the security policy script. (*Id.*)

C. Ground 2: Claim 15 is Obvious Over Farino, Richman, Wiegel, and Moore

1. **Claim 15: “The method of claim 11 further comprising representing, through a visual editor, each normalized physical sensor as an object containing one or more attributes that define a spatial relationship to other objects through one or more defined rules, the visual editor including a drag and drop graphical user interface component that facilitates creation of rules through interconnection of rule objects and definition of rule properties.”**

As fully explained for claim 11[f], Figure 3 of Wiegel depicts policy builder pane **320** (excerpted at right) that is used to

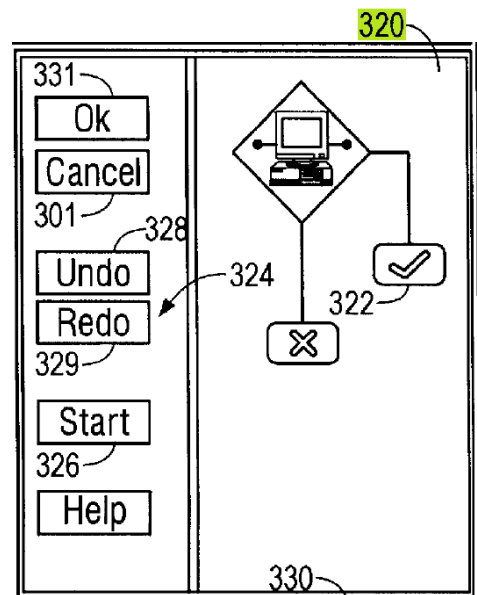
construct a security policy. (Wiegel, 16:2-8.)

“The user constructs a security policy by arranging

one or more icons **322** into a representation of the policy in the policy builder pane **320.**” (Wiegel,

16:5-8.) “As the security policy is constructed, the administration component **206** generates a source

script that defines the policy in a scripting language, and displays the script in the script pane **330.**” (Wiegel, 16:9-12.) The **“visual editor”** under the proposed



combination thus corresponds to the user interface allowing the administrator to create security policies for physical security and networking resources of a site. This visual editor includes, but is not limited to, policy builder pane 320 discussed above – the “**visual editor**” under the proposed combination also includes surrounding on-screen user interface features shown in Figures 3 and 9 that further allow the administrator to apply created security policies to physical security and network devices shown in the network tree. (Madisetti, ¶147; Figs. 3 & 9.)

The proposed Farino-Richman-Wiegel combination, therefore, “**represent[s], through a visual editor, each normalized physical sensor as an object.**” This is because under the proposed combination as explained for claim 11[f], the visual editor would have allowed the administrator to also represent physical security sensors (in addition to computer networks) as visual icons, for example, as shown in the network tree in Wiegel. (Wiegel, Fig. 3 (network tree 314), Fig. 9 (network tree 720).) The physical sensors are “**normalized**” because their sensor data and other communications are mapped to a common XML data representation format in order to incorporate the sensor into the centralized security system, as explained in detail for claims 11[b] and [d]. (Madisetti, ¶¶79-81, 89-96.)

These objects represented through the visual editor also “**contain[] one or more attributes that define a spatial relationship to other objects through one**

or more defined rules.” These attributes include information defining how and where objects will appear in relationship to each other.

Turning back Wiegel Figure 9, it shows what happens after dragging security policy **319** onto an icon in the network tree:

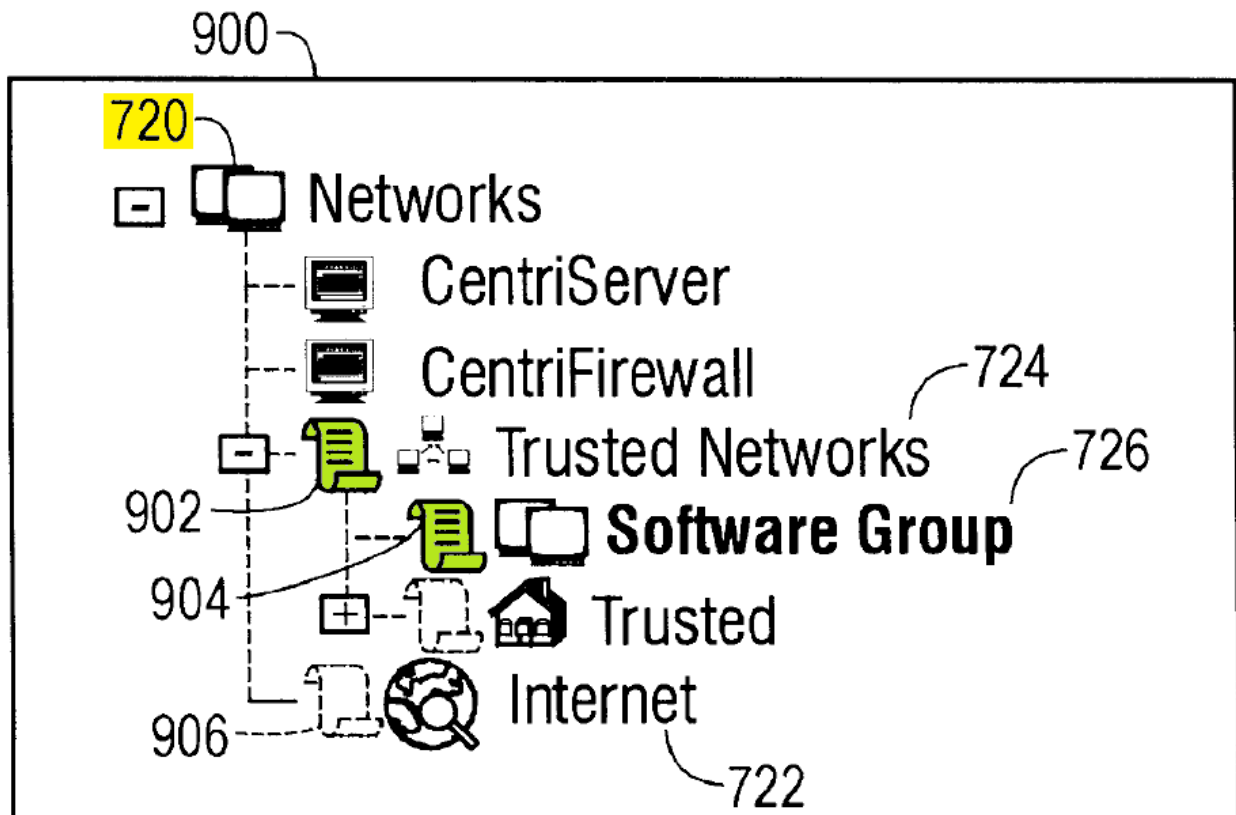


FIG. 9

(Wiegel, Fig. 9 (highlighting added), 27:34-46.) It would have been obvious that objects as shown in Figure 9 contain “**one or more attributes that define a spatial relationship to other objects,**” because the objects are arranged in a hierarchical tree view based on role/location within the network. As explained for claim 11[f], Wiegel does not disclose how the system internally organizes resources as displayed

in network tree 720, but it would have been obvious that the system would have included data (attributes) to record how these resources relate to one another within the hierarchy (*e.g.*, which resources are children of other resources in the hierarchical tree), to visually depict this relationship to the user. (Madisetti, ¶¶149, 123.)

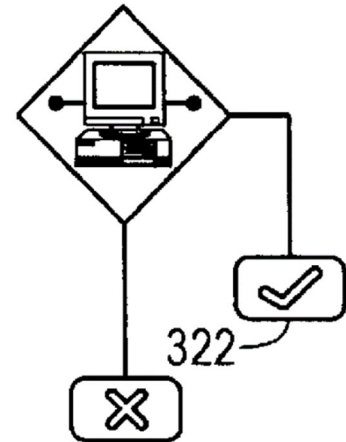
Networks tree 720 also shows policy icons 902 and 904 (shown in green) to indicate that security policies have been applied to the Trusted Networks object 724 and Software Group object 726, respectively, using the drag-and-drop process. (Wiegel, 27:43-46.) Objects 724 and 726, as explained for claim 11[f], have further attributes defining a spatial relationship to other on-screen objects, in this case, the security policies represented as 902 and 904. (Madisetti, ¶¶122-124, 149.) The spatial relationship of these objects – as with any set of objects displayed by any computer program – is defined “**through one or more defined rules**,” *i.e.*, the logic in the software that decides where and how to display them. The “**defined rules**” in this case specify how and where to display the hierarchically organized objects and, where applicable, their adjacent security policy objects. (Madisetti, ¶149.)

The claim continues that “**the visual editor include[es] a drag and drop graphical user interface component that facilitates creation of rules through interconnection of rule objects and definition of rule properties.**” The proposed Farino-Richman-Wiegel combination renders obvious all aspects of this language with the exception of “**a drag and drop graphical user interface component.**”

Petitioner will thus address that portion of claim 15 separately below in the further combination with **Moore** (EX1006).

With respect to “**facilitat[ing] creation of rules through interconnection of rule objects and definition of rule properties,**” the rule objects correspond to the interconnected icons in the policy builder (shown at right).

As explained for claim 11[f], these rule objects include diamond-shaped “condition symbols” (such as the service symbol shown at right), as well as “action symbols” (shown as the X and checkmark shown at right). (Madisetti, ¶¶115-



116; Wiegel, 16:40-56 (describing condition symbols), 16:57-65 (describing action symbols), Fig. 5 (showing more complex exemplary security policy).)

According to Wiegel, “[w]hen a currently displayed symbol of a policy is clicked with the right mouse button, the administration component **206** displays a pop-up menu that presents choices that are currently available for changing the symbol’s associated data, for changing the symbol type, and for extending policy logic from the symbol.” (Wiegel, 19:10-14.) Figure 4C presents an example:

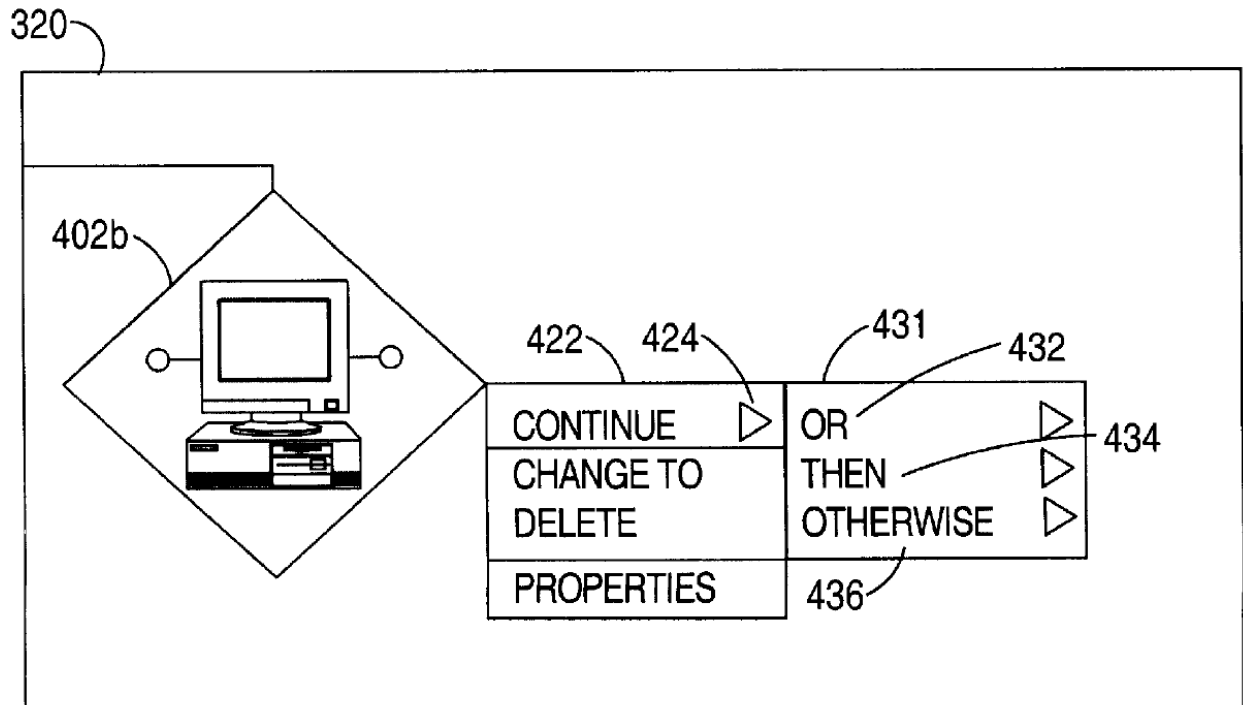


FIG. 4C

(Wiegel, Fig. 4C.) Figure 4C shows a pop-up menu **422** displayed in response to the user right-clicking and selecting condition symbol **402b**. (Wiegel, 19:19-25.) In this example, the user selected the Continue command **424** from the menu, and thus, was presented with the OR command **432**, THEN command **434**, and OTHERWISE command **436**. (Wiegel, 19:34-38.) “The OR, THEN, and OTHERWISE commands express logical conditions that can be used to join a condition symbol to another symbol.” (Wiegel, 19:38-41.)

For example, the “**OR**” command **432** can be used to create and select a new condition for the security policy, which will be linked to the selection condition symbol **402b**. (Wiegel, 19:55-20:6.) When this occurs, “the administration component creates a new condition symbol and displays the new symbol in the

policy builder pane **320** at a position adjacent to the symbol that is being extended, linked with a horizontal line.” (Wiegel, 20:7-11, 20:11-22 & Fig. 5 (showing condition symbol **502** joined to condition symbol **504** via horizontal line **510**)).) The “**THEN**” command **434**, as its name implies, can be used to specify the action to take place when the selected condition is true. (Wiegel, 20:27-31.) For example, the user can specify “Accept,” “Reject,” or “User Next Policy” (Fig. 4E (**446**, **448**, **450**)), thus creating a new action symbol and linking that new symbol to the condition symbol. (Wiegel, 20:46-59.)

The proposed combination thus discloses and renders obvious “**facilitat[ing] creation of rules** [*i.e.* security policy rules] **through interconnection of rule objects** [*i.e.*, interconnection of condition and action symbols] **and definition of rule properties** [*i.e.*, by defining the conditional logic].” (Madisetti, ¶¶152-154.)

The only aspect of claim 15 not disclosed by the Farino-Richman-Wiegel combination is the requirement that creation of rules and definition of rule properties be facilitated by “**a drag and drop graphical user interface component.**” These functions are instead facilitated through pop-up menus as discussed. But it would have been obvious, in further view of **Moore** (EX1006), to adapt the user interface to use drag and drop, as shown below.

“Drag and drop” refers to a well-known user interface gesture in which a user can select an object (for example using a computer mouse or other pointing device)

to drag it to another location on the screen. (Madisetti, ¶156.) **Moore** (EX1006) discloses a user interface for allowing users to assign properties (such as metadata properties) to data items using a drag and drop gesture. (Moore, Abstract, ¶0035.)

Moore describes his system by reference to a familiar user interface for organizing digital music, as shown in Figure 6:

FIG. 6

Filter by...	Song Title	Time	Album	Genre	Rating	Count
Music Links ▾	• Scooter	3:06	Music For the Morning	Alternative Country	☆☆☆	1
☐ All Songs	• Sleep Alone	6:34	Great Expectations Soun...	Alternative Country	☆☆☆☆	5
☐ Online Store	• Metro Area	8:23	Great Expectations Soun...	Alternative Country	☆☆	6
☒ iPod	• Hollywood	1:21	Great Expectations Soun...	Electronic	☆☆☆☆☆	2
⌚ Recently played	• If I were you	2:32	Great Expectations Soun...	Electronic	☆☆☆	3
☒ ☆ Rating	• Mad World	6:56	A Bugged out Mix	Electronic	☆	23
☒ 📁 Playlists	• Silk Road	3:06	A Bugged out Mix	Electronic	☆☆☆☆	7
☒ 🎤 Artists — 221	• Dust	6:34	A Bugged out Mix	Electronic	☆☆	10
○ Madonna — 643	• Encounter	8:23	Time for a Dance	Pop	☆☆☆☆☆	8
○ 50 Cent — 645	• Forest	1:21	Come Away with Me	Pop	☆☆☆☆	50
○ Chocolate — 647	• Hippies I h...	2:32	Come Away with Me	Rock	☆☆☆☆	10
○ Kruder & Dorfmei... — 649	• Spread YW...	6:56	Come Away with Me	Rock	☆☆	23
○ Eminem — 651	• Down to Up...	3:06	Come Away with Me	Rock	☆☆☆☆☆	50
☒ 🎵 Genre	• Top	1:21	Come Away with Me	Rock	☆☆☆☆	10
☒ 📄 View 203	• Rough Jump	1:21	Come Away with Me	Rock	☆☆	23
	• Run Around	2:32	Come Away with Me	Rock	☆☆☆☆	50
	• Tweet	6:56	Great Expectations Soun...	Soundtrack	☆☆	3
	• Just You	3:06	Great Expectations Soun...	Soundtrack	☆☆☆☆☆	53

(Moore, Fig. 6, ¶0022 (highlighting added).) The user interface shown in Figure 6 would have been familiar to almost anyone who had ever used software for organizing digital music such as Apple iTunes. The left column shows a property tree 203 with a number of nodes including Artists 221, which in this case has been expanded 641 to show five artists 643, 645, 647, 649, and 651. (Moore, ¶0026.) The right of property tree 203 shows a list of individual songs with each column providing specific information about each song (e.g., “Song Title,” “Time,” etc.).

Petitioner has highlighted an exemplary artist 649 (“Kruder & Dorfmeister”) in yellow, and an exemplary song (“Encounter”) in green, for illustration. Moore provides an example in which the user assigns the song “Encounter” to the artist “Kruder & Dorfmeister,” using drag and drop:

FIG. 19

Filter by...	Song Title	Time	Album	Genre	Rating	Count
Music Links	• Scooter	3:06	Music For the Morning	Alternative Country	☆☆☆	1
• All Songs	• Sleep Alone	6:34	Great Expectations Soun...	Alternative Country	☆☆☆☆	5
• Online Store	• Metro Area	8:23	Great Expectations Soun...	Alternative Country	☆☆	6
• iPod	• Hollywood	1:21	Great Expectations Soun...	Electronic	☆☆☆☆☆	2
• Recently played	• If I were you	2:32	Great Expectations Soun...	Electronic	☆☆☆	3
• Rating	• Mad World	6:56	A Bugged out Mix	Electronic	☆	23
• Playlists	• Silk Road	3:06	A Bugged out Mix	Electronic	☆☆☆	7 1871
• Artists — 221	• Dust	6:34	A Bugged out Mix	Electronic	☆☆	10
• 50 Cent	• Encounter	8:23	Time for a Dance	Pop	☆☆☆☆☆	8
• Chocolate	• Forest	1:21	Come Away with Me	Pop	☆☆☆	50
• Kruder & Dorfmeister	• Hippies I h...	2:32	Come Away with Me	Rock	☆☆☆☆	10
• Eminem	• Top	1:21	Come Away with Me	Rock	☆☆☆☆	23
• Genre	• Rough Jump	1:21	Come Away with Me	Rock	☆☆	50
• View	• Run Around	2:32	Come Away with Me	Rock	☆☆	10
	• Tweet	6:56	Great Expectations Soun...	Soundtrack	☆☆	23
	• Just You	3:06	Great Expectations Soun...	Soundtrack	☆☆☆☆	50
					☆☆	3
					☆☆☆☆	53

(Moore, Fig. 19 (highlighting added).) “In this case the user is dropping the item 1871 onto the Artists value ‘Kruder & Dorfmeister’ 649, and the checkbox 1093 associated with that value is highlighted. This is thus another mechanism by which a user can assign metadata to items.” (Moore, ¶0035.) The selected song remains in the item list after this operation – the drag and drop operation merely assigned the “Kruder & Dorfmeister” property to that song. (*Id.*; Madisetti, ¶¶157-158.)

Moore describes his system by reference to organizing digital music, but makes clear that the technique can be applied to any system in which “data items are

organized and queried by way of metadata properties.” (Moore, ¶0010.) The techniques of Moore would have been applicable to other types of data items, including rule objects under the proposed combination. (Madisetti, ¶159.)

Manner, Rationale, and Motivation to Combine: It would have been obvious to apply the user interface teachings of Moore to the Farino-Richman-Wiegel system, predictably resulting in the security system under the proposed combination in which the user interface was further adapted to include **“a drag and drop user interface component that facilitates creation of rules through interconnection of rule objects and definition of rule properties.”** (Madisetti, ¶160.) Moore is an analogous reference in the field of computer user interfaces and would have been reasonably pertinent to problems facing the inventors. (Madisetti, ¶164.)

Under the Farino-Richman-Wiegel combination, as discussed, the **“interconnection of rule objects and the definition of rule properties”** occurs using pop-up menus in Wiegel that allow the user to create interconnected condition and/or action symbols and define the conditional logic for the security policy. It would have been obvious to adapt this user interface to allow the user to make these types of selections using a drag and drop user interface component. (Madisetti, ¶161.) The drag and drop user interface in Moore would have been particularly suitable for Wiegel because it allows for the use of drag and drop to make selections within a hierarchical list of properties, and thus, would have been readily adaptable

to the proposed combination. (Madisetti, ¶162; Moore, ¶0035, Figs. 18-19.) The adaptability is further enhanced by the fact that Moore and Wiegel both employ the common “tree” user interface control.

An ordinarily skilled artisan would have been amply motivated to combine. Drag and drop is one of a finite number of well-known user interface tools in computer graphical user interfaces. One of the key benefits of drag and drop is that it mimics how human beings manipulate physical objects in the real world, and is thus exceptionally easy to learn and use. (Madisetti, ¶¶156, 165.)

Drag and drop also has advantages over other techniques such as the pop-up menus used in Wiegel. The pop-up menus **422** in Wiegel require the user to select (e.g. by right-clicking) the condition symbol and then navigate through one or more pop-up menus to make the desired selection, which can be cumbersome. But with drag and drop, the user can simply grab the condition symbol and drop it onto an appropriate on-screen object representing the desired selection. An ordinarily skilled artisan would have understood that, for many users, the drag and drop technique would have been preferable. (Madisetti, ¶166.)

Drag and drop would also have enabled the visual editor to provide greater consistency in the location of on-screen controls. This is because pop-up menus **422** in Wiegel appear and expand out from the location of the selected condition symbol. (Wiegel, Fig. 4C.) Pop-up menus **422** will thus appear in different places on the

Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

screen depending on the location of the selected condition symbol, which could have been jarring to users and caused them to lose focus. But with drag and drop, the available property selections could have been placed in a fixed place (such as a property tree as in Moore), thus providing a more consistent user interface for the assignment of properties. (Madisetti, ¶167.) An ordinarily skilled artisan would have had at least a reasonable expectation of success. (Madisetti, ¶168.)

VII. CONCLUSION

Petitioner respectfully requests IPR institution.

Dated: December 17, 2022

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: / Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,968 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: December 17, 2022

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 8,234,704 B2**, including all exhibits (**Nos. 1001-1012**) and related documents, are being served via Priority Mail Express on the 17th day of December, 2022, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

111078 - Staniford Tomita LLP
P.O. Box 591838
San Francisco, CA 94159

And, via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the District of Delaware entitled *HID Global Corp. v. Vector Flow, Inc.*, No. 21-1769 (GBW) (D. Del.) as follows:

Scott D. Sherwin
MORGAN, LEWIS & BOCKIUS LLP
110 North Wacker Drive
Chicago, IL 60606-1511

DATED: December 17, 2022

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
COOLEY LLP
1299 Pennsylvania Ave. NW,
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001