

Declaration of Vijay K. Madiseti, Ph.D. in Support of
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 No. 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

DECLARATION OF VIJAY K. MADISETTI, PH.D.

Table of Contents

	Page
I. INTRODUCTION AND QUALIFICATIONS	7
A. Qualifications and Experience	7
B. Materials Considered	14
II. PERSON OF ORDINARY SKILL IN THE ART	15
III. STATEMENT OF LEGAL PRINCIPLES	18
A. Claim Construction	18
B. Obviousness	20
IV. THE '704 PATENT	26
A. Overview of the Specification	26
B. The Challenged Claims	30
V. CLAIM CONSTRUCTION	30
VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS	31
A. Brief Summary of Prior Art	32
1. Farino [EX1003]	32
2. Richman [EX1004]	34
3. Wiegel [EX1005]	38
4. Moore [EX1006]	44
B. Ground 1: Obviousness of Claims 11-14 Over Farino, Richman, and Wiegel	48
1. Independent Claim 11	48
(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])	48

Table of Contents
(continued)

	Page
(b) “accepting sensor data from each security sensor 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]).....	58
(c) “defining individual user profiles and their respective access privileges and credentials in the system;” (Claim 11[c]).....	68
(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]).....	70
(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]).....	76

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]).....	83
(1)	“defining physical security policies of the site in the context of user profiles at all sites	84
(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”	86
(3)	“...that are applied to the actionable events normalized to the common data representation format to produce normalized event data; and” (Claim 11[f])	101
(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]).....	103

Table of Contents
(continued)

	Page
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.”	106
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.”	108
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.”	113
C. Ground 2: Obviousness of Claim 15 Over Farino, Richman, Wiegel and Moore	115
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.”	115
VII. NO SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS	131

Table of Contents
(continued)

Page

VIII. CONCLUSION	135
------------------------	-----

I, Vijay K. Madiseti, Ph.D., declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

1. I received my Bachelor of Technology (Honors) in Electronics and Electrical Communication Engineering at the Indian Institute of Technology (IIT) in Kharagpur, India in 1984. I obtained my Ph.D. in Electrical Engineering and Computer Science at the University of California, Berkeley, in 1989. I received the Demetri Angelakos Outstanding Graduate Student Award from the University of California, Berkeley and the IEEE/ACM Ira M. Kay Memorial Paper Prize in 1989.

2. I am a tenured Professor in the College of Computing and Electrical and Computer Engineering at the Georgia Institute of Technology (“Georgia Tech”). I am knowledgeable and familiar with wireless communications, microprocessor architecture, hardware, RF, cellular networks, ASIC design, computer engineering, embedded systems, digital signal processing, and associated software and firmware design for wireless and telecommunications terminals and base stations. I have created and taught undergraduate and graduate courses in hardware and software design for signal processing and wireless communication circuits at Georgia Tech for the past twenty years. Additionally, I have been active in the areas of wireless communications, digital signal processing, integrated circuit design (analog & digital), software engineering, system-level design methodologies and tools, and

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

software systems. I have been the principal investigator (“PI”) or co-PI in several active research programs in these areas, including DARPA's Rapid Prototyping of Application Specific Signal Processors, the State of Georgia's Yamacraw Initiative, the United States Army's Federated Sensors Laboratory Program, and the United States Air Force Electronics Parts Obsolescence Initiative. I have received an IBM Faculty Award and NSF's Research Initiation Award.

3. I have designed several specialized computer and communication systems over the past two decades at Georgia Tech for tasks such as wireless audio and video processing and protocol processing for portable platforms, such as cell phones and PDAs. I have worked on designing systems that are efficient from performance, size, weight, area, and thermal considerations. I have developed courses and classes for the industry on these topics, and many of my lectures in advanced computer system design, developed under the sponsorship of the United States Department of Defense in the late 1990s, are available for educational use at "<http://www.eda.org/rassp>" and have been used by several U.S. and international universities as part of their course work. Some of my recent publications in the area of design of wireless communications systems and associated protocols are listed in Exhibit A. I graduated more than 20 Ph.D. students that now work as professors or in technical positions around the world.

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

4. I have been an active consultant to industry and various research laboratories (including Massachusetts Institute of Technology Lincoln Labs and Johns Hopkins University Applied Physics Laboratory). My consulting work for MIT Lincoln Labs involved high resolution imaging for defense applications, where I worked in the area of prototyping complex and specialized computing systems. My consulting work for the Johns Hopkins Applied Physics Lab (“APL”) mainly involved localization of objects in image fields, where I worked on identifying targets in video and other sensor fields and identifying computer architectures and circuits for power and space-efficient designs.

5. I have founded three companies in the areas of embedded software, military chipsets involving imaging technology, and wireless communications. I have supervised the Ph.D. dissertations of over twenty engineers in the areas of computer engineering, signal processing, communications, rapid prototyping, and system-level design methodology, of which five have resulted in thesis prizes or paper awards. The first of the companies I founded, VP Technologies, offers products in the area of semiconductor integrated circuits, including building computing systems for imaging systems for avionics electronics for the United States Air Force and the US Navy, since 1995. I remain a director of VP Technologies. The second of these companies, Soft Networks, LLC, offers software

for multimedia and wireless computing platforms, including the development of a set-top box for Intel that decodes MPEG-2 video streams, wireless protocol stacks, and imaging codecs for multimedia phones. The technology involved with the design, development, and implementation of the Intel set-top box included parsing the bit streams, decoding communications protocols, extracting image and video data, and then processing for subsequent display or storage. The third of these companies, Elastic Video, uses region of interest based video encoding or decoding for capturing high quality video at very low bit rates, with primary application for wireless video systems.

6. I have authored more than sixty refereed journal publications and around forty peer reviewed conference publications. I have been active in research in the area of wireless and mobile communications and some of my recent peer-reviewed publications in this area include: (i) Mustafa Turkboylari & Vijay K. Madiseti, Effect of Handoff Delay on the System Performance of TDMA Cellular Systems, Proceedings of the Fourth IEEE Conference on Mobile and Wireless Communications Network 411-15 (Sept. 9-11, 2002); (ii) Loran A. Jatunov & Vijay K. Madiseti, Computationally-Efficient SNR Estimation for Bandlimited Wideband CDMA Systems, 5 IEEE Transactions on Wireless Communications, no. 12 (2006) at 3480-91; and (iii) Nimish Radio, Ying Zhang, Mallik Tatipamula & Vijay K.

Declaration of Vijay K. Madisetti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

Madisetti, Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions, 100 Proceedings of the IEEE, no. 4 (April 2012) at 841-54. I have extensive experience analyzing, designing, and testing systems based on 3GPP Technical Specifications, including specifications describing WCDMA and HSDPA technologies. I have been active in the area of location-based services and wireless localization techniques since the mid-1990s, and have authored several papers on location-based services, including, Vijay K. Madisetti et al., Mobile Fleet Application Using SOAP and System on Devices (SyD) Middleware Technologies, Communications, Internet, and Information Technology (2002) at 426-31. I have served as associate editor or on the editorial board for technical journals, including IEEE Transactions on Circuits & Systems II, International Journal in Computer Simulation, and International Journal in VLSI Signal Processing.

7. I have authored or co-authored several books, including VLSI Digital Signal Processors (IEEE Press 1995) and the Digital Signal Processing Handbook (CRC Press, 1998, 2010). I co-authored Quick-Turnaround ASIC Design in VHDL (Kluwer Academic Press 1996) and Platform-Centric Approach to System-on-Chip (SoC) Design (Springer 2004). I am also the editor of several books, including the three-volume DSP Handbook set: Volume 1: Digital Signal Processing Fundamentals, Volume 2: Video, Speech, and Audio Signal Processing and

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

Associated Standards, and Volume 3: Wireless, Networking, Radar, Sensory Array Processing, and Nonlinear Signal Processing, published in 2010 by CRC Press, Boca Raton, Florida. More recently I have authored Cloud Computing (2014, CreateSpace Press), and Internet of Things (2014, CreateSpace), and the book, Cloud Computing, was nominated as a Notable Book of 2014 by the Association of Computing Machinery (ACM) in July 2015.

8. I also have expertise in a technology commonly known as the Internet of Things (IoT), which refers to a set of technologies that allow physical “things” beyond traditional computers (*i.e.*, things that are not typically associated with Internet connectivity) to exchange and transfer data over computer networks such as the Internet. I co-wrote a book on the subject, *Internet of Things: A Hands-On Approach* (2014), ISBN 978-0996025515. IoT applications are used with a wide variety of things such as thermostats, irrigation controls, environmental sensors, intrusion detection sensors, smoke detection and gas detectors, and many other devices, all of which are specifically discussed in the above-mentioned book. I thus have specific expertise in technologies for receiving and processing sensor data from network-connected sensors.

9. I have been elected a Fellow of the IEEE, for contributions to embedded computing systems. The Fellow is the highest grade of membership of the IEEE, a

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

world professional body consisting of over 300,000 electrical and electronics engineers, with only one-tenth of one percent (0.1%) of the IEEE membership being elected to the Fellow grade each year. Election to Fellow is based upon votes cast by existing Fellows in IEEE. I have also been awarded the 2006 Frederick Emmons Terman Medal by the American Society of Engineering Education for contributions to Electrical Engineering, including authoring a widely used textbook in the design of VLSI digital signal processors. I was awarded VHDL International Best PhD Dissertation Advisor Award in 1997 and the NSF RI Award in 1990. I was Technical Program Chair for both the IEEE MASCOTS in 1994 and the IEEE Workshop on Parallel and Distributed Simulation in 1990. In 1989 I was recognized with the Ira Kay IEEE/ACM Best Paper Award for Best Paper presented at the IEEE Annual Simulation Symposium.

10. I have submitted approximately thirty invention disclosures and provisional patents over the past ten years. None of them are patents yet.

11. I have testified as an expert witness before. Over the past six years, I've testified as an expert in more than 20 proceedings. About half of the proceedings in which I have testified as an expert were in the area of 2G/3G/4G wireless receiver design, including six to seven in the area of digital physical layer design.

12. I have attached a more detailed list of my qualifications as **Exhibit A**.

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

13. I am being compensated for my time working on this matter at my standard hourly rate plus expenses. I do not have any personal or financial stake or interest in the outcome of the present proceeding, and the compensation is not dependent on the outcome of this IPR and in no way affects the substance of my statements in this Declaration.

B. Materials Considered

14. The analysis that I provide in this Declaration is based on my education and experience in the field of computer systems, as well as the documents I have considered, including U.S. Patent No. 8,234,704 B2 (“’704 patent”) [EX1001] and its prosecution history. The ’704 patent states on its face that it issued from an application filed on September August 14, 2007, and claims priority to a provisional application filed on August 14, 2006. For purposes of this Declaration, I have assumed August 14, 2006 as the effective filing date for the ’704 patent. I have cited to or considered the following documents in my analysis below:

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”)
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”)

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

Exhibit No.	Description of Document
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)
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)

II. PERSON OF ORDINARY SKILL IN THE ART

15. I understand that an assessment of claims of the ’704 patent should be undertaken from the perspective of a person of ordinary skill in the art as of the

earliest claimed priority date, which I have assumed to be August 14, 2006. I have also been advised that to determine the appropriate level of a person having ordinary skill in the art, the following factors may be considered: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor.

16. In my opinion, a person of ordinary skill in the art as of August 2006 would have possessed a bachelor's degree in software engineering, computer science, computer engineering, or electrical engineering with at least two years of experience in distributed or network-based computer systems. A person could also have qualified as a person of ordinary skill in the art with some combination of (1) more formal education (such as a master's of science degree) and less technical experience or (2) less formal education and more technical or professional experience in the fields listed above. Acquired as part of the person's basic computer education and/or experience, as the '704 patent makes clear, a person of ordinary skill in the art would have had a working knowledge about various ways to receive data from network-connected devices (such as sensors) and then processing and/or acting upon that data as needed.

17. I note that my formulation of a person of ordinary skill in the art is consistent with the formulation adopted by Patent Owner. In particular, I have reviewed a document entitled “Declaration of Mark Jones in Support of Plaintiff HID Global Corporation’s Opening Claim Construction Brief” (EX1011), which I understand was served by the patent owner in pending litigation between Petitioner and Patent Owner. In this document, Dr. Mark Jones, an expert retained by Patent Owner, opines that a person of ordinary skill in the art for purposes of the ’704 patent “would have been an individual who has completed a Bachelor of Science degree in Computer Science, or equivalent field, and had approximately three years of professional experience in the design or development of distributed computer systems (although, for example, more professional experience may compensate for less formal education, and vice versa).” (EX1011, ¶26.) For purposes of my application of the prior art in this Declaration, the formulation proposed by Dr. Jones is not materially different from the one I apply here. My opinions set forth in this Declaration, therefore, would remain the same regardless of whether I adopted my own formulation or the one articulated by Dr. Jones.

18. My opinions regarding the level of ordinary skill in the art are based on, among other things, my more than 20 years of experience in computer science and engineering, my understanding of the basic qualifications that would be relevant

to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, and employees, both past and present.

19. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above (under either my formulation or the one provided by Patent Owner), my analysis and opinions regarding the '704 patent have been based on the perspective of a person of ordinary skill in the art as of August 2006.

III. STATEMENT OF LEGAL PRINCIPLES

A. Claim Construction

20. The following sets forth my understanding of the legal principles of claim construction from counsel. I understand that under the legal principles, claim terms are generally given their ordinary and customary meaning, which is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, *i.e.*, as of the effective filing date of the patent application. I further understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which a claim term appears, but in the context of the entire patent, including the specification.

21. I am informed by counsel that the patent specification, under the legal principles, has been described as the single best guide to the meaning of a claim

term, and is thus highly relevant to the interpretation of claim terms. And I understand for terms that do not have a customary meaning within the art, the specification usually supplies the best context of understanding the meaning of those terms.

22. I am further informed by counsel that other claims of the patent in question, both asserted and unasserted, can be valuable sources of information as to the meaning of a claim term. Because the claim terms are normally used consistently throughout the patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims. Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.

23. I understand that the prosecution history can further inform the meaning of the claim language by demonstrating how the inventors understood the invention and whether the inventors limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be. Extrinsic evidence may also be consulted in construing the claim terms, such as my expert testimony.

24. I have been informed by counsel that, in *Inter Partes* Review (IPR) proceedings, a claim of a patent shall be construed using the same claim construction standard that would be used to construe the claim in a civil action filed in a U.S. district court (which I understand is called the “*Phillips*” claim construction

standard), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

25. I have been instructed by counsel to apply the “*Phillips*” claim construction standard for purposes of interpreting the claims in this proceeding, to the extent they require an explicit construction. The description of the legal principles set forth above thus provides my understanding of the “*Phillips*” standard as provided to me by counsel.

26. I understand that there are a number of claim construction disputes between Patent Owner and Petitioner, as detailed in the Declaration of Mark Jones in Support of Plaintiff HID Global Corporation’s Opening Claim Construction Brief (EX1011), which I understand was served by Patent Owner in pending litigation between Petitioner and Patent Owner. For purposes of applying the prior art identified in this Declaration, however, it is unnecessary for me to provide express claim constructions. This is because, as shown below, the prior art renders obvious the challenged claims under even the narrowest constructions proposed in the pending litigation, or under any other reasonable construction.

B. Obviousness

27. The following sets forth my understanding of the legal principles of

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

obviousness from counsel. I understand that a patent claim is obvious if, as of the critical date (i.e., either the earliest claimed priority date (pre-AIA) or the effective filing date (AIA)), it would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs.

28. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other facts known as “secondary considerations” such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus,” with the invention claimed in the patent at issue.

29. I understand that a reference qualifies as prior art for obviousness purposes when it is analogous to the claimed invention. The test for determining what art is analogous is: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the inventor’s endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

30. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense.

31. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid.

32. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

solution encompassed by the patent's claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together like the pieces of a puzzle. I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this finite number of predictable solutions leads to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does not more than yield predictable results. Thus, if a person of

ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

33. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. I understand that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem.

34. I understand the following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other market forces if the variations would have been predictable to a person of ordinary skill in the art;
- (3) Combining prior art elements according to known methods to yield predictable results;
- (4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;
- (5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);
- (6) Simple substitution of one known element for another to obtain predictable results; or
- (7) Use of a known technique to improve similar products, devices, or methods in the same way.

35. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether one of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the

field of endeavor, would have seen a benefit to selecting the combination claimed.

36. I understand that the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. The test for obviousness, in other words, is not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the teachings of the prior art as a whole.

IV. THE '704 PATENT

A. Overview of the Specification

37. 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.) The '704 patent acknowledges that physical security systems that provide access control and security monitoring were known and commonplace. ('704, 1:29-46 (e.g., "Almost all businesses, large commercial and government facilities, and many homes utilize physical security systems....").) The '704 patent states that a variety of vendors manufacture different and proprietary physical security systems, and because a large entity, such as a big corporation, may use multiple systems, the creation and enforcement of uniform security policies across these systems can be challenging. ('704, 1:53-2:2.) The

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

'704 patent accordingly asserts that there is a need for an integrated system that uses a uniform data format to integrate disparate physical security systems, as well as a visual tool for defining security policies that are enforced across the disparate systems. ('704, 2:11-24.)

38. The '704 patent states that it discloses “a system and method for the integration of data and events from physical access control and security monitoring systems that are normalized to a standardized definition for enforcement of standardized rules, created through a visual policy editor, affecting persistence, propagation of data and generation of alerts and notifications for physical security, network and IT systems.” ('704, 3:14-21.) Figure 1A, reproduced below, illustrates a computer network system **100** that implements the security system:

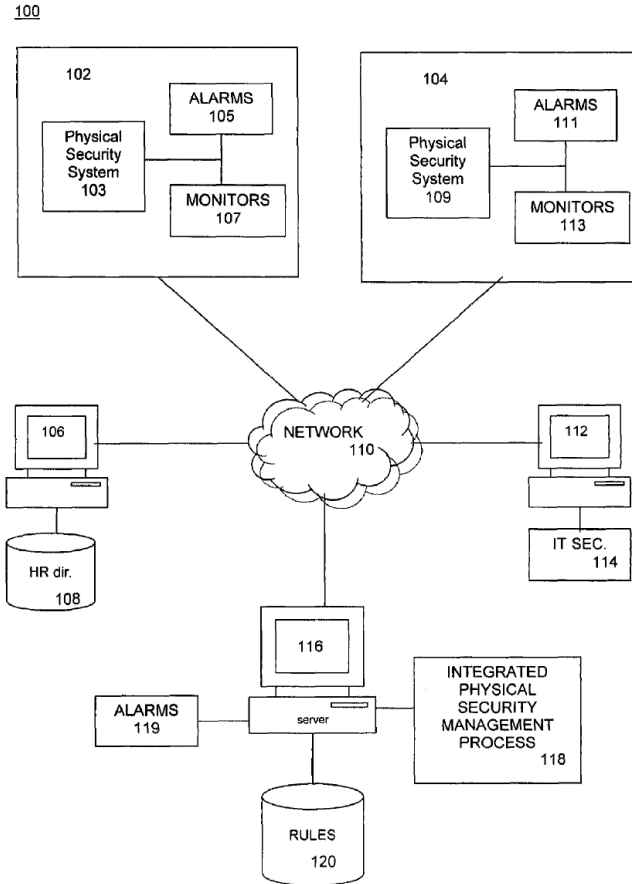


FIG. 1A

(’704, Fig. 1A.) As shown above, the 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), that each include 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

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

process **118**, which “generally comprises hardware and/or software components to achieve integration, normalization, rules creation and processing of physical security systems data and events in different facilities, such as facilities **102** and **104**.” (’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.) A visual policy editor, illustrated in Fig. 7 below, facilitates creation of security policies and related rules:

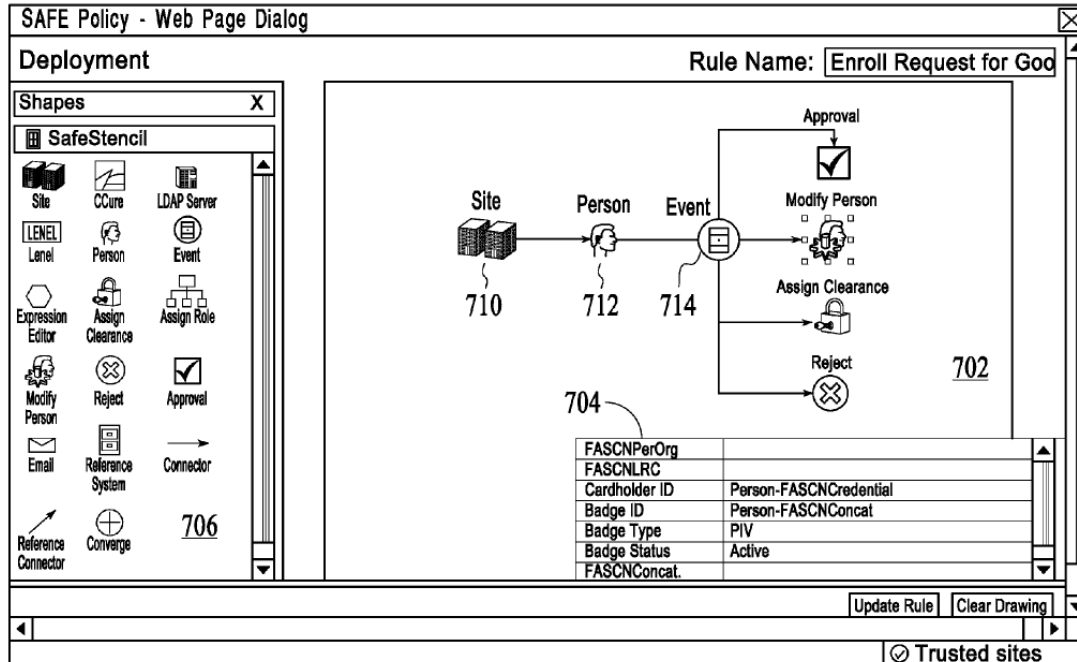


FIG.7 700

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

B. The Challenged Claims

39. This Declaration addresses claims 11-15. I address the claims further in my detailed analysis in **Part VI** below.

V. CLAIM CONSTRUCTION

40. As I noted above, I am informed that in the underlying district court litigation between Petitioner and Patent Owner, the claim construction process is currently ongoing. I understand that Petitioner and Patent Owner have exchanged

claim construction briefs for certain terms in dispute (EX1010-1011), but that the district court has issued no order resolving those disputes. For purposes of applying the prior art to the claims in the present *inter partes* review, however, it is unnecessary to resolve the parties' disputes and I have not performed an analysis to evaluate the parties' competing positions. My analysis below will show how the prior art satisfies each limitation under any of the constructions advanced in the pending litigation (or any other reasonable construction). To the extent Patent Owner attempts to distinguish my mapping of the prior art to the claims based on claim interpretations that differ (either explicitly or implicitly) from the ones it advocated for in the underlying litigation (as identified above), I reserve the right to respond accordingly.

VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS

41. I have reviewed and analyzed the prior art references and materials listed in **Part I.B** above. In my opinion, the claims of the '704 patent are rendered obvious based on the following prior art:

Ground	Basis for Challenge Under §103(a)	Claim(s)
1	Farino (EX1003), in view of Richman (EX1004) and Wiegel (EX1005)	11-14
2	Farino, in view of Richman, Wiegel and Moore (EX1006)	15

42. I am informed by counsel that each reference listed above qualifies as

prior art to the challenged claims because each reference was filed and/or published before the earliest claimed priority date for the '704 patent. I have also cited additional references (EX1007-1010), for information that would have fallen with the knowledge of a person of ordinary skill in the art. These references were also filed and/or published before the earliest priority date for the '704 patent.

A. Brief Summary of Prior Art

1. Farino [EX1003]

43. Farino, entitled “Unified Network and Physical Premises Access Control Server,” discloses an access control system for managing both network and physical security. (Farino, Abstract.) I am informed that Farino qualifies as prior art because it was filed on October 26, 2005 and published on April 26, 2007, both dates being before the filing date of the application of the '704 patent.

44. My Declaration cites Farino as a primary reference that discloses the majority of the limitations of the challenged claims. Farino describes a unified access control server that is “unified” in that it provides a centralized security system for both network security (e.g. security of computers and computer networks) and physical security (e.g. security of physical facilities). As Farino explains:

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. Advantageously, server **200** unifies credential verification and associated policies and policy enforcement for physical facilities and network-enabled devices. This unification facilitates implementation of new security policies that strengthens both physical and network security.

(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 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.)

45. To ensure physical security, the system in Farino can interface with many different types of physical security sensors such card readers:

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; other exemplary sensor types include “fire and smoke sensors” (¶0079).

46. Farino further 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 (“Access control gateway **125** allows legacy ACDs and DCPs that were previously deployed in and around the facility to be used with the present invention without further modification to the physical access control system.”).) I will describe Farino in more detail below.

2. Richman [EX1004]

47. Richman, entitled “Method and Protocol for Real Time Security System,” discloses a security system for responding to events from physical security sensors. (Richman, Abstract.) I am informed that Richman qualifies as prior art

because it was filed on June 20, 2002 and published on November 6, 2003, both dates being before the earliest priority date for the '704 patent.

48. Richman discloses a security system that is similar in many respects to Farino. My Declaration cites Richman primarily for limitations in the challenged claims reciting that sensor data from security sensors must be mapped to “**a common file representation format.**” As explained previously, Farino 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.) But Farino does not describe the operation of the access control gateway **125** in detail, and as such, I have relied on Farino in further combination with Richman.

49. 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 addresses this issue 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 “conversion modules” that perform a function similar to the access control gateway **125** in Farino – they receive sensor data in its native form translating it into another format, but in the case of Richman, into standardized 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. (Richman, ¶¶0030-0041.) As Richman notes, “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.)

50. Richman provides several concrete examples of converting device-specific data from security devices into data in standardized XML format. Richman provides one example in which a tenant seeks to access a room secured with a keypad and enters a username and password. (Richman, ¶¶0120-0121.) Richman converts

the received data from the keypad into the following standardized XML format:

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

51. As another example, Richman discloses an example in which a tenant seeks access to a room secured with a fingerprint or eye detection system. (Richman, ¶¶0128-0129.) The conversion module, upon receiving the sensor data, converts it into a 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 translating sensor data from an intrusion or motion detection sensor into XML format to indicate that a window has been broken. (Richman, ¶¶0114-0119.) I will provide more detail about Richman in my discussion below.

3. Wiegel [EX1005]

52. Wiegel, entitled “Graphical Network Security Policy Management,” describes a user interface for creating and managing security policies in a networked computer system. (Wiegel, Abstract.) I am informed that Wiegel qualifies as prior art because it was filed on December 11, 1998 and issued on November 19, 2002, both dates being before the earliest priority date for the ’704 patent.

53. My Declaration cites Wiegel primarily for claim 11[f], which recites

certain visual user interface features for creating security policies, which are not expressly disclosed in Farino or Richman. Wiegel explains that, in the prior art, creating a security policy involved the complex task of preparing an ordered set of rules, which often required that learn “text-based, command-line interfaces.” (Wiegel, 1:62-65.) “The cryptic command sets and low-level knowledge about networks and network protocols required to program rule-based security policies,” Wiegel explains, “are complicated for administrators to learn.” (Wiegel, 3:38-41.)

54. Wiegel purports to address this problem by providing a graphical user interface for creating and managing security policies. An example of Wiegel’s user interface is shown in Figure 3:

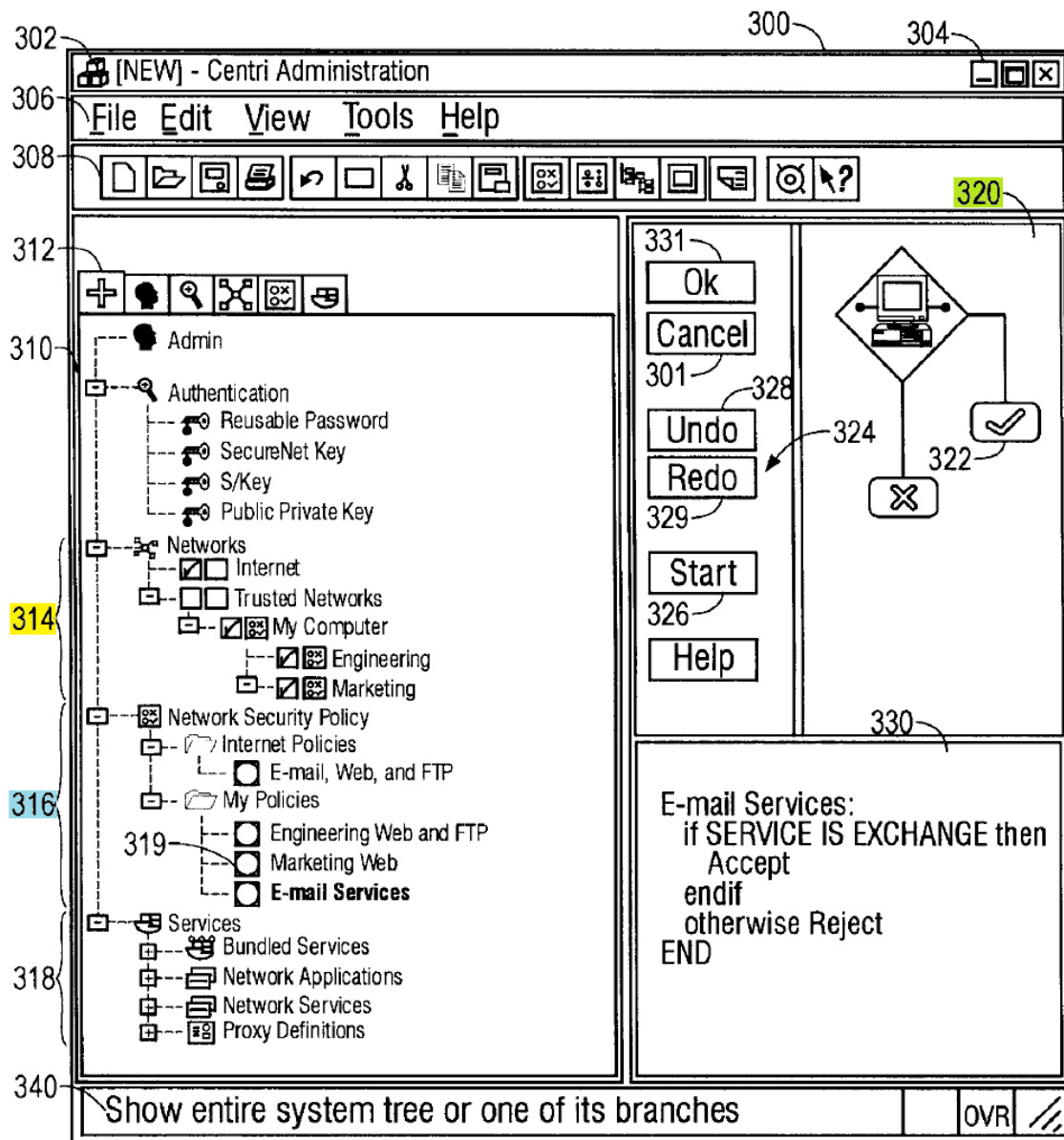


FIG. 3

(Wiegel, Fig. 3 (highlighting added).) Figure 3 above 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:

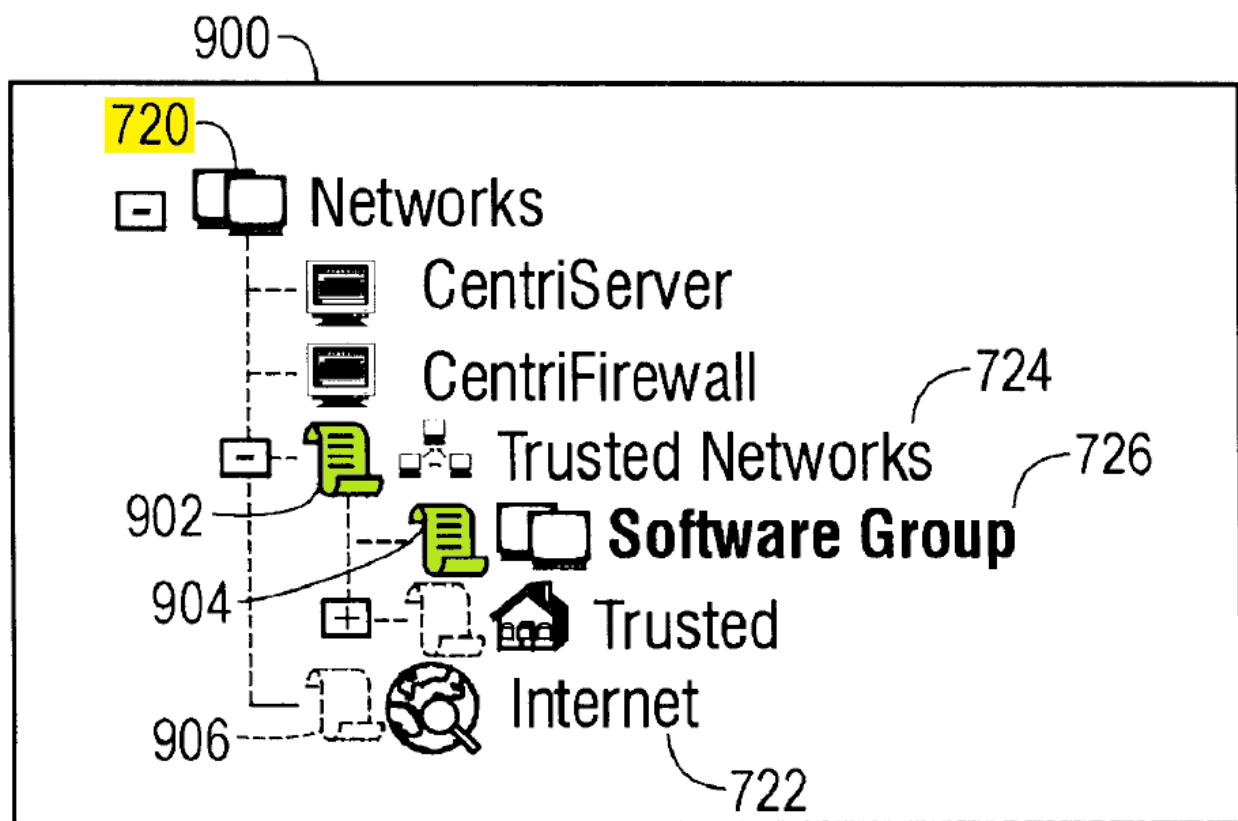
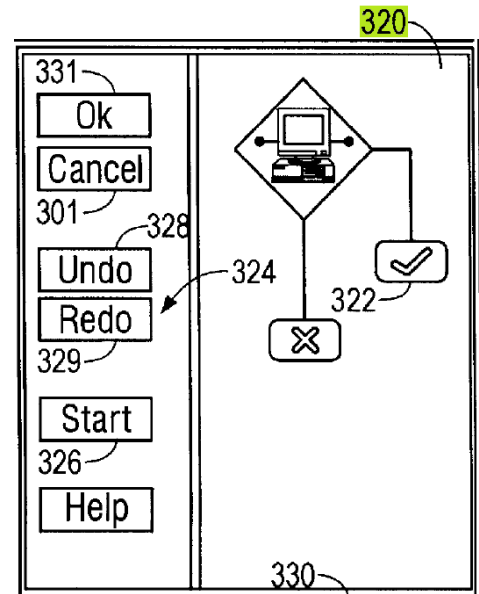


FIG. 9

(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.)

55. 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.



56. 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:

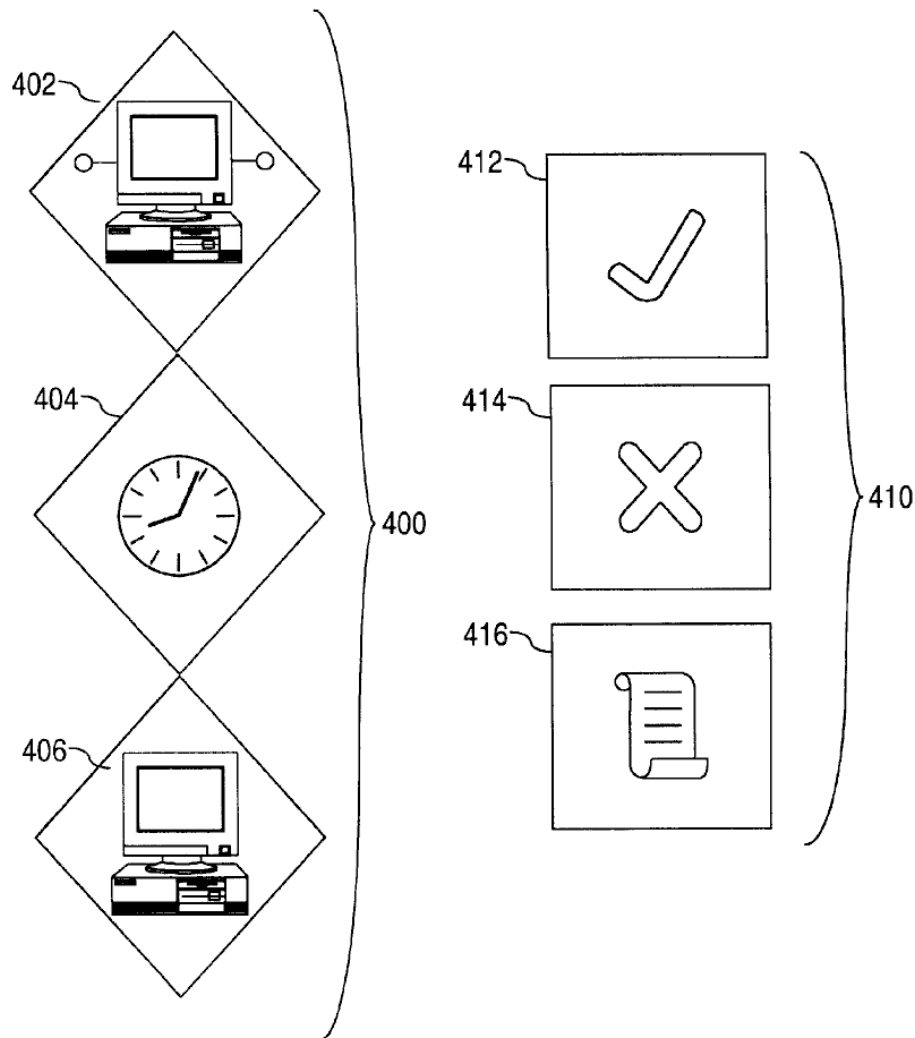


FIG. 4A

(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.) The service condition symbol **402** “represents one of the services in the services tree **318** [Fig. 3].” (Wiegel, 16:43-45.) The time range symbol **404** (shown as the clock) is used to construct policies to reject

information (packets) based on the time of day when they arrive or are sent. (Wiegel, 16:47-50.) The destination symbol **406**, shown at the bottom-left, “represents an end point at which data packets are sent.” (Wiegel, 16:52-56.) Other condition symbols may be provided. (Wiegel, 16:55-56.)

57. 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.) I will provide more detail about Wiegel in my discussion below.

4. Moore [EX1006]

58. Moore, entitled “Property Tree for Metadata Navigation and Assignment,” describes user interface techniques for assigning properties to data items. (Moore, Abstract.) I am informed that Moore qualifies as prior art because it was filed on April 30, 2004 and published on November 3, 2005, both dates being before the earliest priority date for the ’704 patent.

59. I have cited Moore solely for **Ground 2** in connection with the “**drag and drop user interface component**” limitation of dependent claim 15. As would have been known to ordinarily skilled artisans, as well as even the most novice

computers users, “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) and “**drag**” it to another location on the screen. The user can then “**drop**” the object by deselecting it (for example by releasing the mouse button), which causes the software to perform an operation. Drag and drop dates back to at least the early Apple Macintosh computers introduced in the mid-1980s, which used drag and drop for common file system operations (*e.g.* moving files between folders, moving files to the trash for deletion, etc.). One of the key benefits of drag and drop is its ease-of-use because it mimics how human beings manipulate physical objects in the real world. It is thus exceptionally easy to learn and use.

60. Moore 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



(Moore, Fig. 2, ¶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 the popular Apple iTunes program). 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,” “Album,” etc.). I have highlighted an exemplary artist 649 (“**Kruder & Dorfmeister**”) in yellow, and song (“**Encounter**”) in green, for illustration.

61. 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
☐ Madonna	Encounter	8:23	Time for a Dance	Pop	☆☆☆☆☆	8
☐ 50 Cent	• Forest	1:21	Come Away with Me	Pop	☆☆☆	50
☐ Chocolate	• Hippies I h...	2:32	Come Away with Me	Rock	☆☆☆☆	10
☒ Kruder & Dorfmeister	• Encounter	8:23	Time for a Dance	Pop	☆☆☆☆☆	8
☐ Eminem	• Top	1:21	Come Away with Me	Rock	☆☆☆☆	50
☐ Genre	• Rough Jump	1:21	Come Away with Me	Rock	☆☆☆	10
☐ View	• Run Around	2:32	Come Away with Me	Rock	☆☆	23
	• Tweet	6:56	Great Expectations Soun...	Soundtrack	☆☆	3
	• Just You	3:06	Great Expectations Soun...	Soundtrack	☆☆☆☆☆	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 item in this case (the song “Encounter”) “does not disappear from the list view; the behavior is more like tagging information to a file.” (Moore, ¶0035.) In other words, the song remains in the item list – the drag and drop operation merely assigned the “Kruder & Dorfmeister” property to that song. Moore thus discloses a simple drag and drop user interface component for assigning properties (e.g., artists) to items (e.g. songs).

62. Although Moore describes his system in the context of organizing digital music, Moore makes clear that the technique is not so limited and can be

applied broadly to any type of computer system in which “data items are organized and queried by way of metadata properties.” (Moore, ¶0010.) The property tree **203** can also be used, for example, to assign properties to “documents, messages, video files, and contacts, with the ProperTree in each case being specifically adapted for the kind of data item that is presented.” (Moore, ¶0022.) In another embodiment, the property tree is a standard control incorporated into an operating system (such as Microsoft Windows), “for application developers to reuse in their own applications.” (Moore, ¶0022.) As I will explained for **Ground 2**, an ordinarily skilled artisan would have understood that the property tree would have been applicable to any type of data items and readily adaptable to the proposed obviousness combination.

B. Ground 1: Obviousness of Claims 11-14 Over Farino, Richman, and Wiegel

1. Independent Claim 11

63. The preamble of claim 11 recites “[a] **method comprising.**” The prior art discloses and renders obvious a 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])**

64. Farino discloses a **“centralized security system”** that includes a

unified access control server. (Farino, ¶0061.) As Farino explains:

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. Advantageously, server 200 unifies credential verification and associated policies and policy enforcement for physical facilities and network-enabled devices. This unification facilitates implementation of new security policies that strengthens both physical and network security.

(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**” as claimed. With respect to the system being “centralized,” the properties of the unified access control server in Farino confirm that the system is centralized. Figure 10 of Farino, for example, specifically shows unified access control server in a centralized configuration:

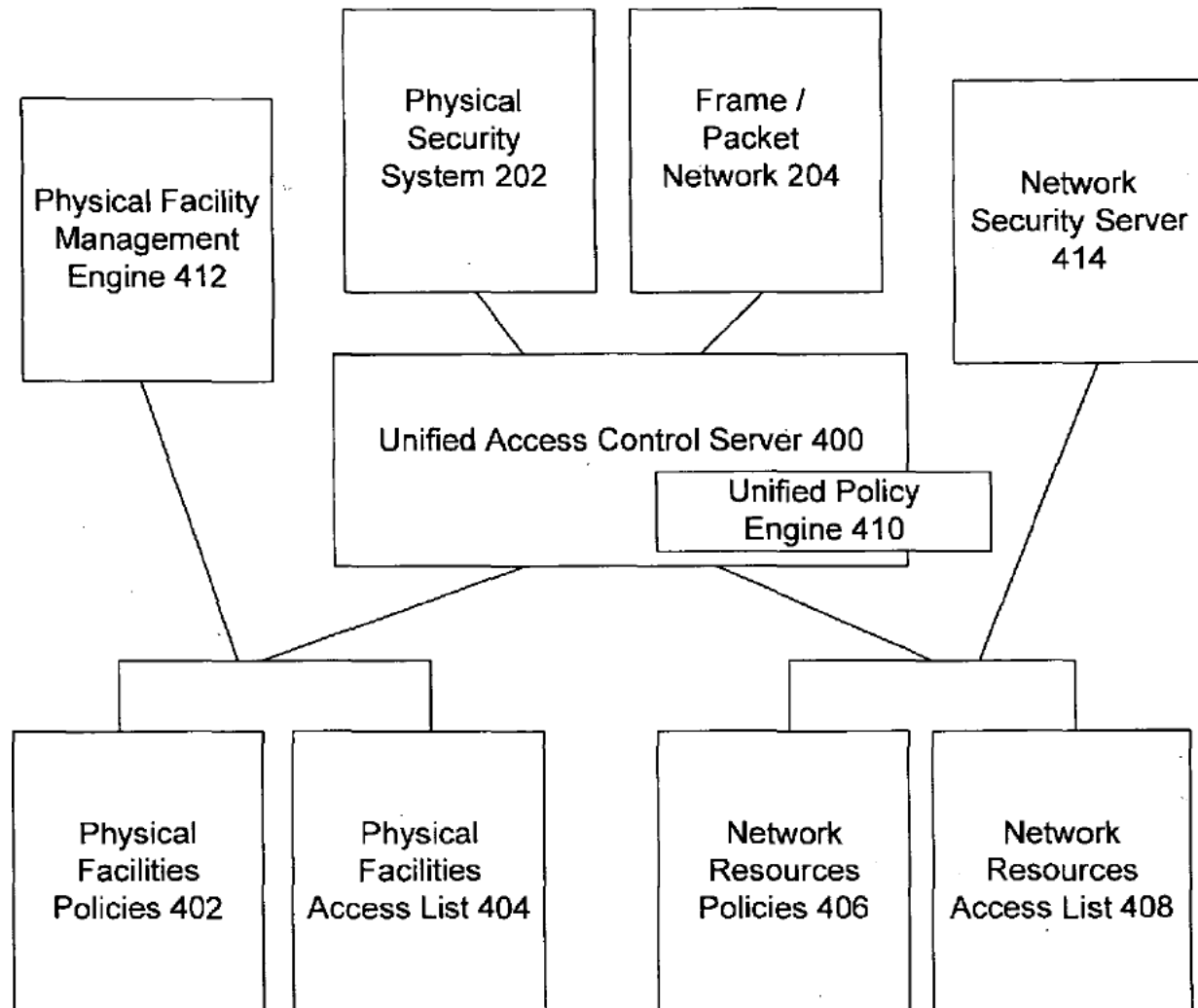


FIGURE 10

(Farino, Fig. 10.) The security system in Farino is centralized because unified access control server provides a unified system for managing physical security across one or more facilities. 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.) A person of ordinary skill would have understood the unified access control server of Farino to facilitate creation of a centralized system because it provides a central hub for physical security and access control. As explained in more detail for claim 11[c] and 11[e] below, moreover, the unified access control server 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.) All of these disclosures confirm that the existence of a “**centralized**” security system, as claimed, because the unified access control server of Farino 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.

65. Farino further discloses “**interfacing in a centralized security system, a plurality of security sensor types distributed throughout a plurality of sites,**” as claimed. The “**plurality of security sensor types**” in Farino includes types of 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 thus discloses a plurality of sensor types.

66. Farino also discloses that the plurality of security sensor types may be **“distributed throughout a plurality of sites,”** *i.e.*, distributed 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.”).) It therefore would have been obvious that the plurality of sensor types of Farino could have been distributed throughout a plurality of different sites, such as different buildings or facilities.

67. Farino also discloses **“interfacing in a centralized security system”** the plurality of sensor types. Farino discloses that security sensors can communicate with the unified access control server, for example, in response to an access control event such as 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 (“In general, unified access control server **200** validates credentials and enforces access policies based on location, type of resource, time of day, duration, or other events, and logs all successful and unsuccessful attempts to access a given resource whether it be on the network or at a physical premises.”), 0065 (“Unified access control server **200** also supports access control monitoring functions and access control events.”).)

68. More specifically, in one embodiment, the sensor types in Farino can interface with the centralized security system, for example, by the unified access control system receiving 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 (“In another embodiment, as shown in the block diagram of FIG. 7B, computer **151** and plurality of ACDs **110** are coupled to a network-connected DCP **229**. DCP **229** aggregates

data from the ACDs, packetizes the data and forwards packets over frame-/packet-based network **28** to unified access control server **200**.”), 0075 (“FIG. 7C illustrates yet another embodiment of the present invention. In this case, each ACD **231** is network-connected and communicates with unified access control server **200** over the frame or packet-based network infrastructure network **228**.”).) As a further disclosure of the “**interfacing**” step, network-connected ACDs are registered with the unified access control server, such as by recording their IP or other network addresses:

Recognizing that network-connected physical access control gateways, or network-connected DCPs or ACDs are pre-requisites for use with the unified access control server, these gateways, DCPs and ACDs will be configured and registered with the unified access control server **200** in any known manner. The unified access control server will record the IP and/or MAC address of the gateway, DCP and associated ACDs as these devices are introduced and connected to the network via well known network protocols such as ARP requests, etc.

(Farino, ¶0077.) Farino therefore discloses the step of “**interfacing**” in unified access control server, a plurality of sensor types distributed throughout a plurality of sites, as recited in the claim.

69. Farino further discloses that “**each sensor type [is] configured to respond to a corresponding type of actionable event.**” This is because each type

of sensor is designed to respond to the particular type of event the sensor was designed to detect. For example, in one embodiment involving ACDs that govern physical access, an actionable event corresponds to a user presenting credentials (such as a security badge) in order to request physical access. (Farino, ¶¶0089 (“The following example policy assumes that a physical access control event will precede a network access control event (i.e. the user will badge-in to a give [sic; given] facility or location and will then proceed to his work area and log-in to the corporate network.”), 0090 (“A user presents credentials at a reader or other ACD.”).) The “**actionable event**” in this example thus 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 ACD. (Farino, ¶¶0082, 0087, 0004-0005.) More specifically, for a card reader ACD, for example, the actionable event 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 password or secret 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). (Farino, *e.g.*, ¶0047 (“Further, 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.”).) 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.

70. Farino also renders obvious that “**each sensor type [is] provided by a different manufacturer of a plurality of manufacturers.**” It would have been obvious to a person of ordinary skill in the art that each sensor type in Farino could have been provided by a different manufacturer or vendor. 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 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 (“Access control

gateway **125** allows legacy ACDs and DCPs that were previously deployed in and around the facility to be used with the present invention without further modification to the physical access control system.”.) Farino thus expressly contemplates that sensor types could have been provided by multiple different manufacturers. It would have been obvious that an organization could have deployed a security system in which each sensor type was provided by a different manufacturer.

71. At a broader level, as with any type of electronic component, 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. For example, **Prokupets** (EX1007), which I have cited to confirm the knowledge of an ordinarily skilled artisan, identifies several examples of vendor-specific specialization of sensor types. 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), among several others. As applied to the prior art, therefore, it would have been obvious that the sensor types in Farino could have come from a variety of different manufacturers, such as (1) security sensor types including door

access controls from a first manufacturer such as Lenel Systems, (2) motion detection/intrusion sensors provided by a second manufacturer such as Digital Monitoring Products, as identified in Prokupets, and so forth. The examples in Prokupets are mere examples; an ordinarily skilled artisan would have appreciated that the sensor types in Farino could have been provided by many other different manufacturers. In any case, it would have been obvious for a security system to have had plurality of sensor types, each sensor type provided by a different manufacturer or a plurality of manufacturers. This would have been a natural result of the specialization within the industry (*e.g.* each manufacturer specializes in a particular sensor type), as explained. 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.

(b) “accepting sensor data from each security sensor 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])

72. I note 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.” I understand that the parties in the underlying litigation have interpreted this language to mean “**accepting sensor data... by an integration module**,” and I have applied that understanding in my analysis.

73. Farino further discloses “**accepting sensor data from each security sensor**,” which takes the form of the data received from the security sensors. In the example of ACDs governing physical access, for example, the sensor data reflects credential information provided by the user at the sensor. (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.”).) The “**sensor data**” thus corresponds to (in this example) 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.

74. Farino further discloses accepting this sensor data by “**an integration module**.” Claim 11 is a method claim, and I understand that Patent Owner and Petitioner have not taken the position that “**integration module**” is a means-plus-function term (under 35 U.S.C. § 112(6)), and as such, I have applied its plain and ordinary meaning. As I will explain below, the claimed integration module is disclosed by the combination of Farino with Richman. The integration module

under the proposed combination takes the form of hardware and software for receiving and processing sensor data from the security sensors.

75. With respect to Farino, as noted, 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, as explained, “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, I have relied on Farino in further combination with Richman (**EX1004**), which I have also relied on for the more detailed requirements of claim 11[d] below.

76. **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 addresses this issue 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. (Richman, ¶¶0030-0041.) As Richman notes, “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.)

77. As I 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 provides one example in which a tenant seeks to access a room secured with a keypad and enters a username and password. (Richman, ¶¶0120-0121.) Richman converts the received data from the keypad into the following standardized XML format:

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

78. As another example, Richman discloses an example in which a tenant seeks access to a room secured with a fingerprint or eye detection system. (Richman, ¶¶0128-0129.) The conversion module, upon receiving the sensor data, converts it into a 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 translating sensor data from an intrusion or motion detection sensor into XML format to indicate that a window has been broken. (Richman, ¶¶0114-0119.)

79. ***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. Under the proposed combination, therefore, 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 the access control gateway **125**, which would have been further adapted to include conversion module functionality of Richman.

80. Accordingly, the integration module under the proposed 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.”).) A person of ordinary skill in the art would have understood the claim term “**agent**” to simply refer to software functionality for mediating between networked components, in this case, the sensor and other components of the Farino system. (*Microsoft Computer Dictionary* (5th ed. 2002) (EX1010), p.22 (“**agent. n...** 3. In client/server applications, a process that mediates between the client and the server.”).) Under the proposed combination, each conversion module agent integrates sensor data from a security sensor type by converting it into standardized XML format for transmission to and processing by the unified access control server in Farino.

81. 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 ACD. (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**,” as claimed, 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.

82. Farino and Richman are closely-analogous references in the same field as the '704 patent of physical security systems, including systems that integrate disparate security systems. ('704, 1:24-27.) Farino and Richman, as explained, disclose similar systems for receiving sensor input from devices in a native data representation and converting it into another format. These references would have been reasonably pertinent to a number of problems facing the inventor, including how to receive and process device-specific sensor data from devices and convert it into another form for processing by a security system.

83. An ordinarily skilled artisan would have been amply motivated to combine. As explained, Farino discloses an access control gateway **125** that can convert native (vendor-specific) sensor data into a format, but Farino does not further describe the conversion or the resulting format. (Farino, ¶¶0040, 0063.) Farino also does not appear to disclose converting native sensor data into an industry *standardized* format; it instead discloses converting into “another access control

system vendor's component formats.” (Farino, ¶0040.) Richman, on the other hand, provides a detailed description of the conversion process and an express motivation to use his system, by disclosing conversion of native (vendor-specific) sensor data into standardized XML format. (Richman, ¶0097.) XML was (and remains) a well-known technique for encoding information, and it was (and remains) widely-used to encode a wide variety of types of data.

84. Richman, in a section entitled “XML advantages,” 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.) Richman thus explains that “it would be highly desirable to have an XML based communications method and protocol capable of enabling the integration of varying security and environmental hardware monitoring devices, and allowing communication between said devices and a core system application for the purpose of monitoring security

systems and/or environmental systems within one or more subject sites, both on site and remotely using direct and indirect means.” (Richman, ¶0045.)

85. The benefits of XML format expressly described in Richman would have been well-known to ordinarily skilled artisans. An ordinarily skilled artisan would have understood the benefit of using XML format because it 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. XML format as discussed in Richman, in other words, would have been a natural choice for Farino for conversion of the native and vendor-specific sensor data provided by the security sensors.

86. An ordinarily skilled artisan would have perceived no technological obstacle to this combination, and would have had every expectation of success. The ability to encode vendor-specific information into XML format was well-known and documented, as Richman itself confirms. (Richman, ¶¶0030, 0040-0041.) The ability to encode sensor data into XML format would have been simple, and in fact trivial, to implement. An ordinarily skilled artisan would have found that the disclosures that I have cited from Farino and Richman provide more than sufficient detail to implement the techniques they describe upon which I rely, without requiring

undue experimentation.

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

87. Farino discloses several ways in which individual user profiles and respective access privileges and credentials are defined within the system. For example, Farino discloses a table or list identifying individual users and their respective access privileges:

The network access control server maintains a network access centric table or list of valid electronic-credentials and an associated list of network resources a given credential holder/user can access based on certain conditions (e.g. policies). It is used to determine authorized network access levels for users or computers attempting to gain network access. 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.

(Farino, ¶0026.) Farino further explains that such a table is also used to define access privileges and credentials for 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.)

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.)

88. Farino thus discloses “**defining individual user profiles and their respective access privileges and credentials in the system,**” as claimed. The defined “**individual user profiles**” in Farino correspond to access-related information (for example in tables or lists) for individual users, who as explained above, can be individually identified by their user names or user identifiers (IDs). (Farino, ¶¶0026, 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 this have included its

own respective access privileges and credentials. The “**access privileges**” in Farino correspond to the facilities or resources the user is permitted to access. (*Microsoft Computer Dictionary* (5th ed. 2002) (EX1010), p.14 (“**access privileges** *n.* The type of operations permitted a given user for a certain system resource on a network or a file server.”).) The “**credentials**” in Farino correspond to the identity establishment credentials such the user identity along with the user’s secret password, fingerprint information, etc. required for access to those facilities or resources. (*Microsoft Computer Dictionary* (5th ed. 2002) (EX1010), p.133 (“**credentials** *n.* A set of information that includes identification and proof of identification that is used to gain access to local and network resources. Examples of credentials are user names and passwords, smart cards, and certificates.”).)

(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])**

89. 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. As explained,

Richman discloses a series of conversion modules that take sensor data from sensors from different manufacturers in native form and convert it into standardized XML format. (Richman, ¶0097.)

90. Richman provides several concrete examples of converting device-specific data from security devices into standardized XML data. Richman provides one example in which a tenant seeking to access a room secured with a keypad enters a username and password. (Richman, ¶¶0120-0121.) Richman converts the received 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 username, password, time and the room number (5) obtained from the keypad. This XML is then transmitted for processing to the SCA. (Richman, ¶¶0122-0124.)

91. Richman discloses another example in which a tenant seeking to access a room secured with a fingerprint or eye detection system. (Richman, ¶¶0128-0129.)

Richman converts the received sensor data 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 XML format data representing from sensors that in response to detecting that a window has been broken. (Richman, ¶¶0114-0119.)

92. The XML format disclosed by Richman discloses a “**common data representation format**” because it provides a common (*e.g.* standardized) format for encoding sensor data. A person of ordinary skill in the art would have recognized XML format as providing an example of a common data representation format. 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”** as claimed.

93. Richman itself repeatedly describes the XML format in which sensor data is converted as a “standardized” format. (Richman, *e.g.*, ¶¶0038 (XML “[p]rovides a standard packaging/transport mechanism for any type of information”), 0066 (“This unique XML based communications protocol is implemented through numerous modules which receive and convert data messages from diverse security devices and sensors, standardize and send converted messages, and encrypt and decrypt said data messages as necessary.”), 0097 (“This [conversion] 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.”), 0098 (“standardized XML message”), 0099 (“standardized message in XML format”).) The fact that the XML format is a **“common data representation format”** is also confirmed by the specific examples above from Richman, which show common XML elements (*e.g.*, “<sensor code>”, “<event code>”, “<attribute code>”) which are used in the XML format by 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.

94. 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 from the XML format examples in Richman. For example, the **“data object”** can take the form of information in the XML messages of Richman identifying the user and/or facility that currently seeks access. The XML examples above 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. 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...”)).

95. With respect to the **“processing information”** included as part of the common data representation format, Richman discloses this limitation in at least two independent 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 must actually process in order to determine if access is permitted (*e.g.*, by matching

password or fingerprint data from the sensor against stored credentials). If access is permitted, for example, the door is opened. (Richman, ¶¶0124-0126; *see also* Farino, ¶¶0018, 0038.)

96. 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. Under the proposed combination, therefore, the common data representation format includes “**a data object and processing information for the sensor data**,” as claimed.

- (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])**

97. As more fully explained for claim 11[c], Farino discloses generating **“access privileges and credentials”** to **“map a defined user profile,”** as claimed. The generation of user profiles containing access privileges and credentials is evidenced, as explained above for 11[c], by access-related information in Farino (for example tables or 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.*, the ability to specify access privileges and credentials for a particular user profile. I will separately address the claim language of claim 11[e] requiring **“unique”** physical access privileges and **“exclusive”** mapping, later in my analysis.

98. 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 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

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 (“Hence, the unified access control server 200 provides all functionality... formerly provided by an access control panel.”), 0062 (“The unified access control server **200** eliminates the need for an Access Control Panel...”), 0071 (“Given a single unified access control server **200** functions as both an ACP and network access control server, it collapses the hierarchy of both systems to minimize the cost of installing and maintaining physical and network access control systems.”)).

privileges and credentials) could have been mapped “**to a spatial hierarchy 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, as explained, it includes such information for each of these security devices.

99. Farino also discloses that the “**physical access credentials maintain a common representation of the user’s identity across the plurality of sites,**” as claimed. Farino discloses a unified access control server that includes an access table, as discussed above. (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.

100. This provides a “**common representation**” because the user identify is stored and used by the unified access control server in Farino which, as explained above, is a centralized system for managing user access credentials and access control policies. Farino expressly describes this list as being a common list:

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.”).)

101. It does not appear that the requirement of a “**common representation**” in claim 11[e] further requires some kind of standardized format for storing access privileges and credentials. But even if the claim were interpreted to impose such a requirement, it would have been obvious based on Richman to store this information in standardized XML format. Richman explains, as noted, that XML “[p]rovides a standard packaging/transport mechanism for any type of information.” (Richman, ¶0038.) Richman provides an example in which individual 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 many well-known benefits of XML outlined in claim 11[b], including standardization and improved interoperability. This would have enabled the unified access control server in Farino to share access privileges and credentials information with other software running on the server through a common and standardized XML format.

102. Finally, the physical access privileges and credentials in Farino

“associate specific user identities with respective actionable events,” as claimed.

As noted above, the common list or table in Farino identifies not only the specific user identities, but also the respective facilities or resources the associated 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.

103. **“Unique” and “Exclusive”**: I note that claim 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 the words **“unique”** and **“exclusively”** could 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.

104. With respect to “**credentials**,” it would have been obvious that a defined user profile could have unique and exclusive credentials. As I explained above for claim 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.

105. 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 persons of ordinary skill in the art and even laypersons, with the concept 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. (*See EX1008* (Vivadelli), ¶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 been given broad privileges for access to all areas within all facilities – a level of access shared by no other user. 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.

- (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])**

106. In light of the length of this limitation, I have broken it into parts.

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

107. Farino also discloses “**defining physical security policies of the site in the context of user profiles at all sites**,” as claimed. I note that there appears to be no antecedent basis for “**the site**,” and as such, I have interpreted claim 11[f] to require defining physical security policies of a site within the system.

108. 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. Here, the “**physical security policies of the site**” was largely covered in my discussion of claim 11[e] above, where I 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.

109. Farino further discloses physical security policies “**of the site in the context of user profiles at all sites,**” as claimed. 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 claim 11[d] above, Farino discloses a centralized and unified access control server that stores and maintains 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 (“Advantageously, server **200** unifies credential verification and associated policies and policy enforcement for physical

facilities and network-enabled devices.”), 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.”).) 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.)

110. Finally, with respect to the act of “**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.) Farino thus discloses the ability to define the claimed policies.

(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**”

111. Neither Farino nor Richman disclose details about the system used to

define or create physical security policies, such as the user interface that allows the administrator to define security policies. I have thus cited 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.)

112. Wiegel explains that, in the prior art, creating a security policy involved the complex task of preparing an ordered set of rules, which often required learning “text-based, command-line interfaces.” (Wiegel, 1:62-65.) “The cryptic command sets and low-level knowledge about networks and network protocols required to program rule-based security policies,” Wiegel explains, “are complicated for administrators to learn.” (Wiegel, 3:38-41.)

113. Wiegel purports to address this problem by providing a graphical user interface for creating and managing security policies. An example of Wiegel’s user interface is shown in Figure 3:

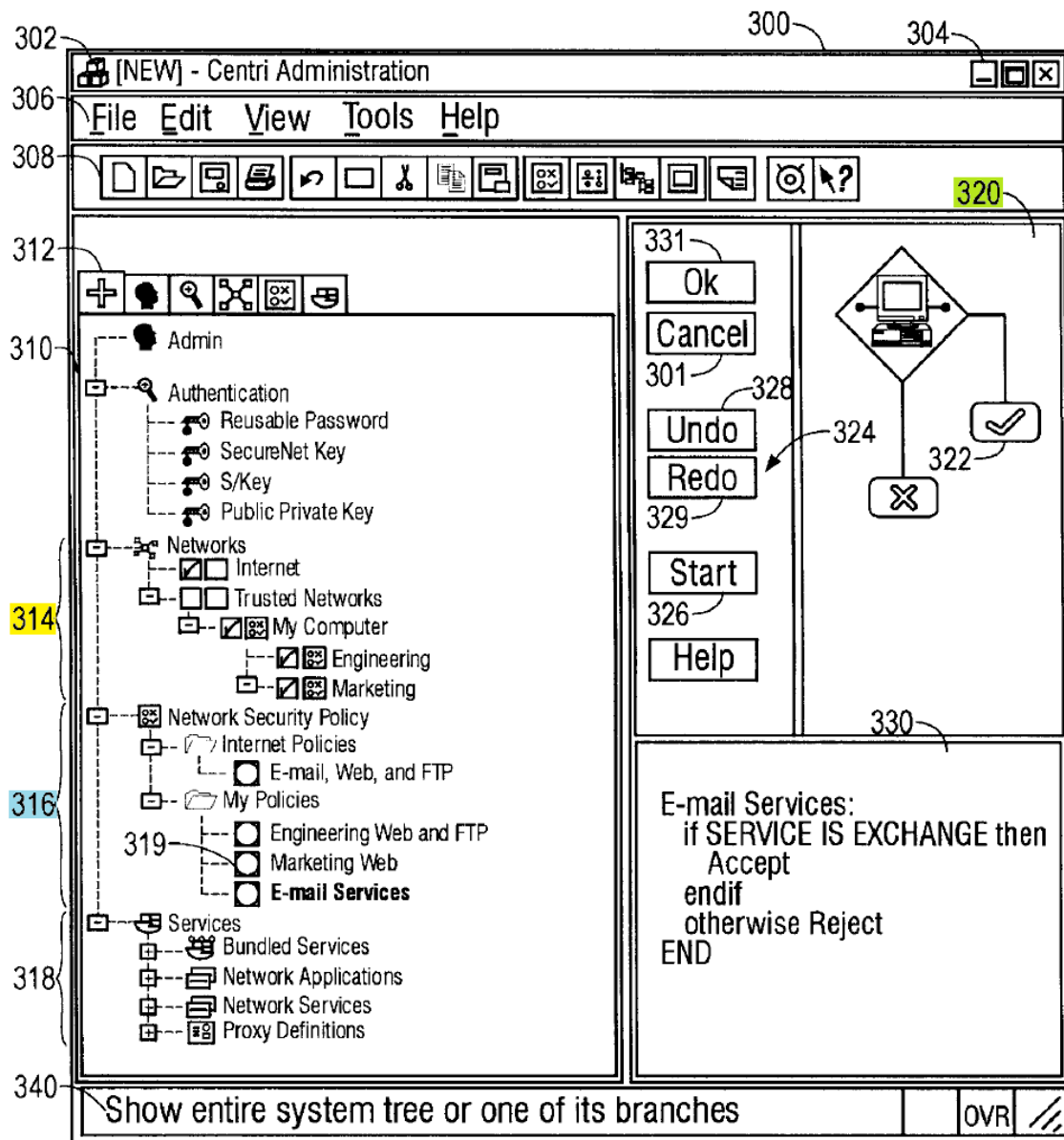


FIG. 3

(Wiegel, Fig. 3 (highlighting added).) Figure 3 above 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:

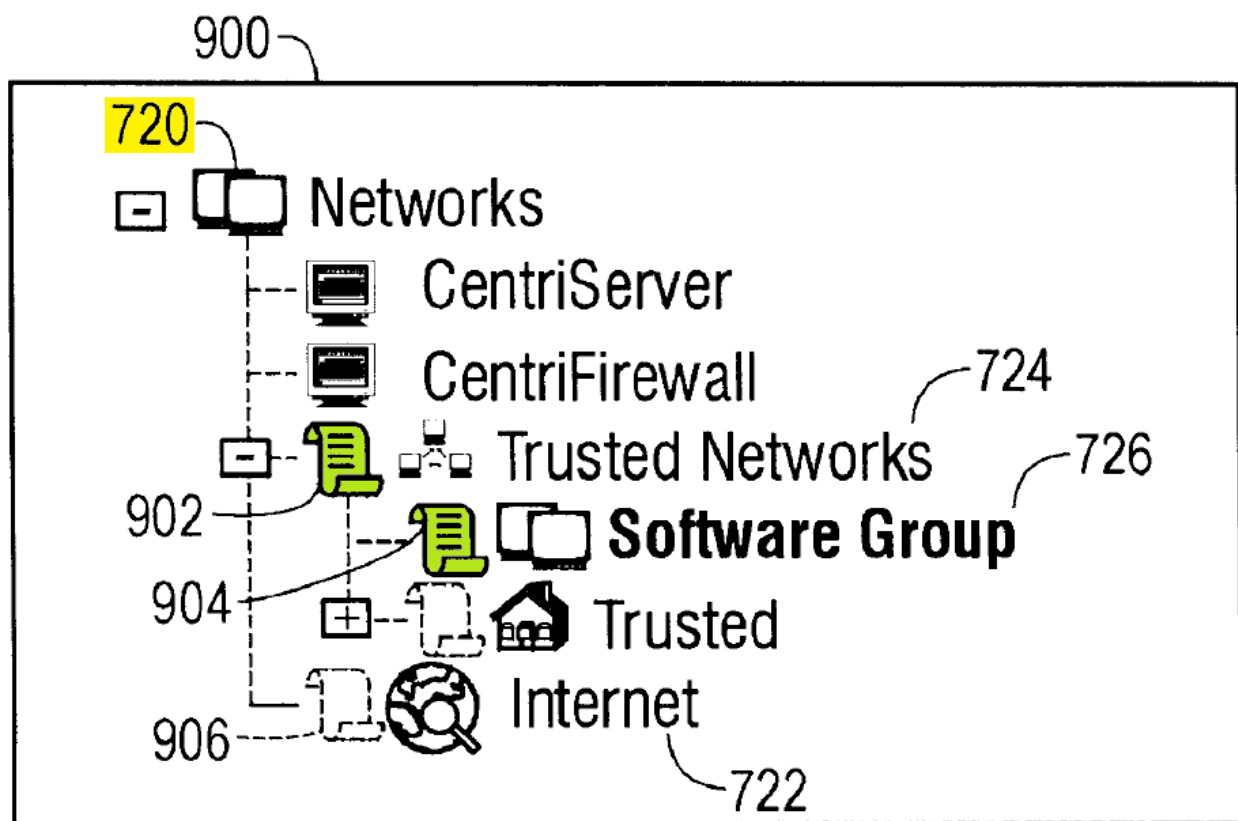


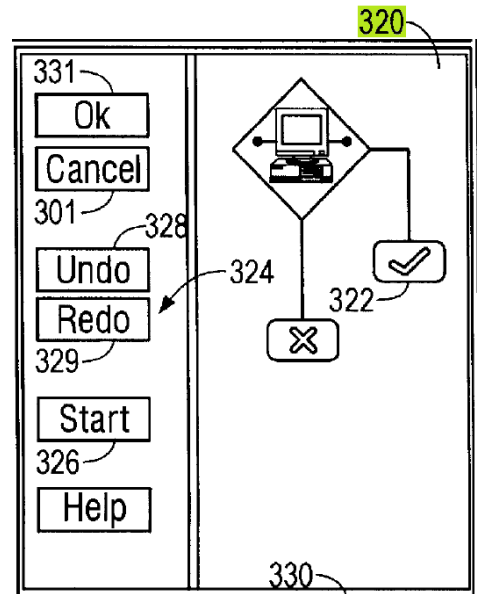
FIG. 9

(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.)

114. 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.

115. 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:

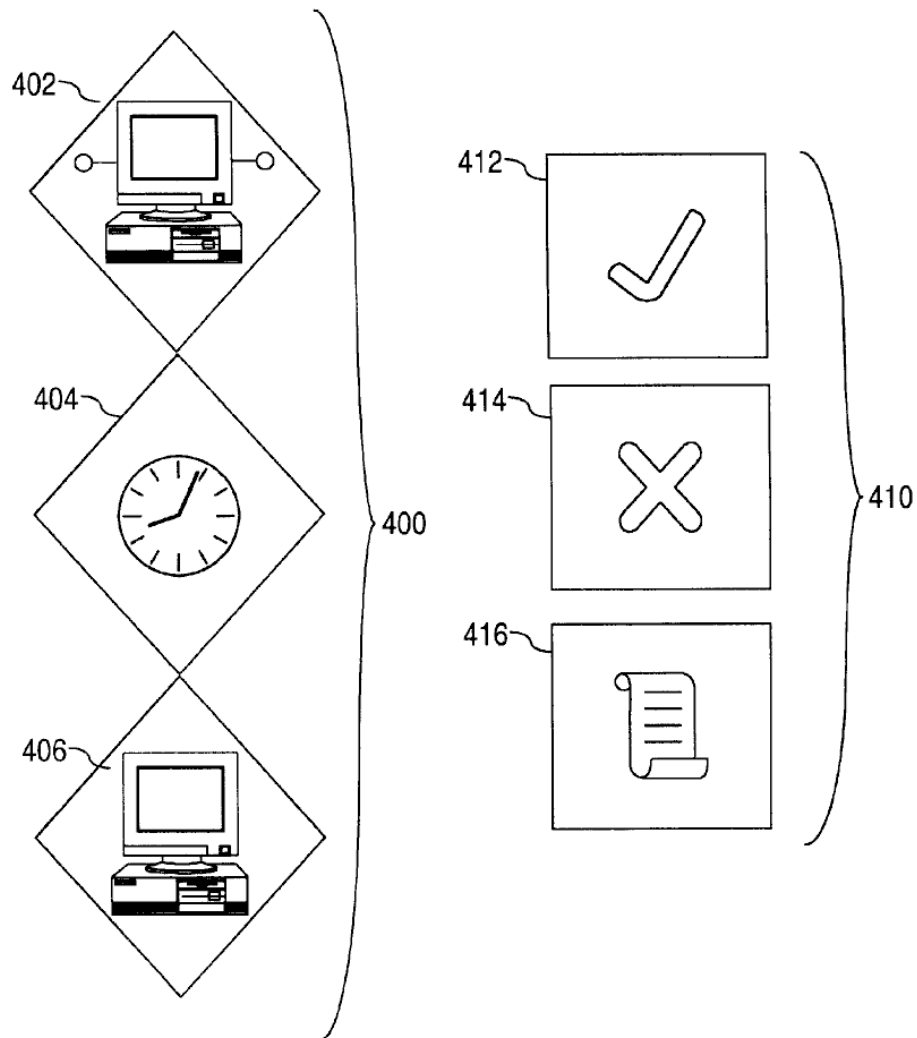


FIG. 4A

(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.) The service condition symbol **402** “represents one of the services in the services tree **318** [Fig. 3].” (Wiegel, 16:43-45.) The time range symbol **404** (shown as the clock) is used to construct policies to reject

information (packets) based on the time of day when they arrive or are sent. (Wiegel, 16:47-50.) The destination symbol **406**, shown at the bottom-left, “represents an end point at which data packets are sent.” (Wiegel, 16:52-56.) Other condition symbols may be provided. (Wiegel, 16:55-56.)

116. 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.)

117. Having outlined the disclosures of Wiegel, I will now explain how Wiegel applies to Farino and Richman. I will first explain the manner of combining the Wiegel to Farino and Richman, and then, describe the motivations to do so.

118. *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 security 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,**” as claimed. Under the proposed combination, the “**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:

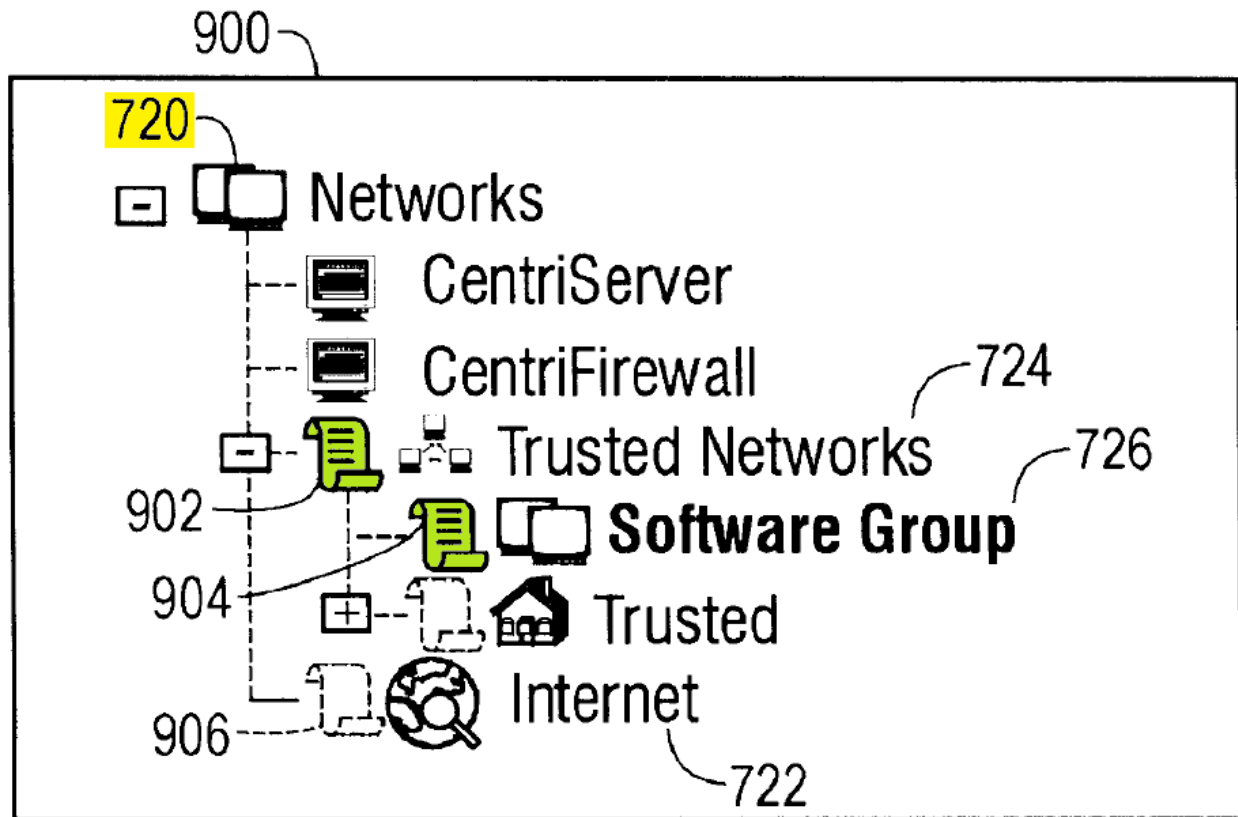


FIG. 9

(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

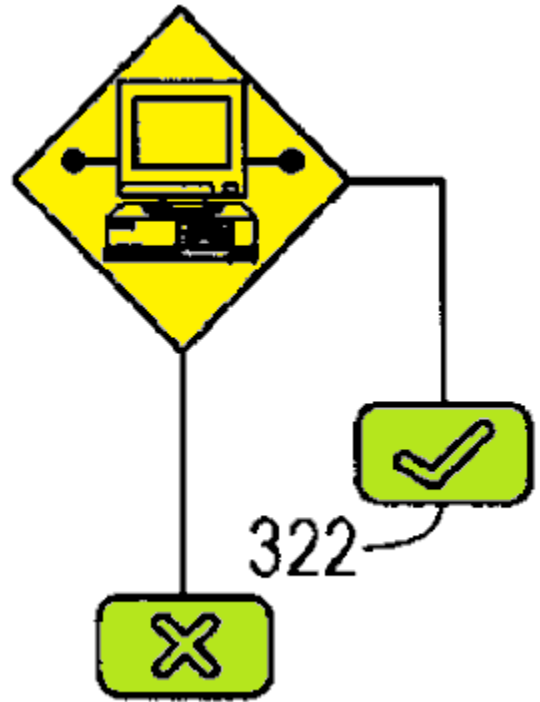
show icons and corresponding textual labels for physical security resources, such as access control devices (ACDs) and the other security devices in Farino discussed above. Under the proposed combination, the icons representing network, IT, and physical resources qualify as “**actionable representations**” because user actions can be taken with respect to them; for example, as explained, the administrator can drag-and-drop a security policy onto an icon to associate that policy with the corresponding resource. (Wiegel, *e.g.*, 14:44-47, 15:38-41, 15:60-63, 27:36-40.)

119. 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,**” as claimed. 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; *see also Microsoft Computer Dictionary* (5th ed 2002), (EX1010), p.467 (“**scripting language. n.** A simple programming language designed to perform special or limited tasks, sometimes associated with a particular application or function.”).) 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. 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.**” 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.

120. 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.

121. I am informed that 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.”* I am further informed that the term **“live objects”** has been interpreted by Petitioner as *“stencil shapes that are used to represent normalized systems, data, and processes.”* I understand that Patent Owner disagrees with these interpretations, but even if adopted, they are disclosed under the proposed combination.



Using the example from the policy builder pane of Figure 3 (shown excerpt at right), the policy in this case was defined using **condition** and **action** symbols (icons) connected to each other in appropriate relationships. These icons also qualify as stencil shapes, as they have standardized and consistent shapes representing systems, data, and processes such as services. The underlying systems, data, and processes are normalized because, as explained, they are created through a graphical user interface with standardized symbols and ultimately expressed in policies written in a common and standard scripting language.

122. These visual rules and live objects also “**contain attributes to define their spatial relationship to the actionable representations,**” as claimed. This is because, as explained, 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:

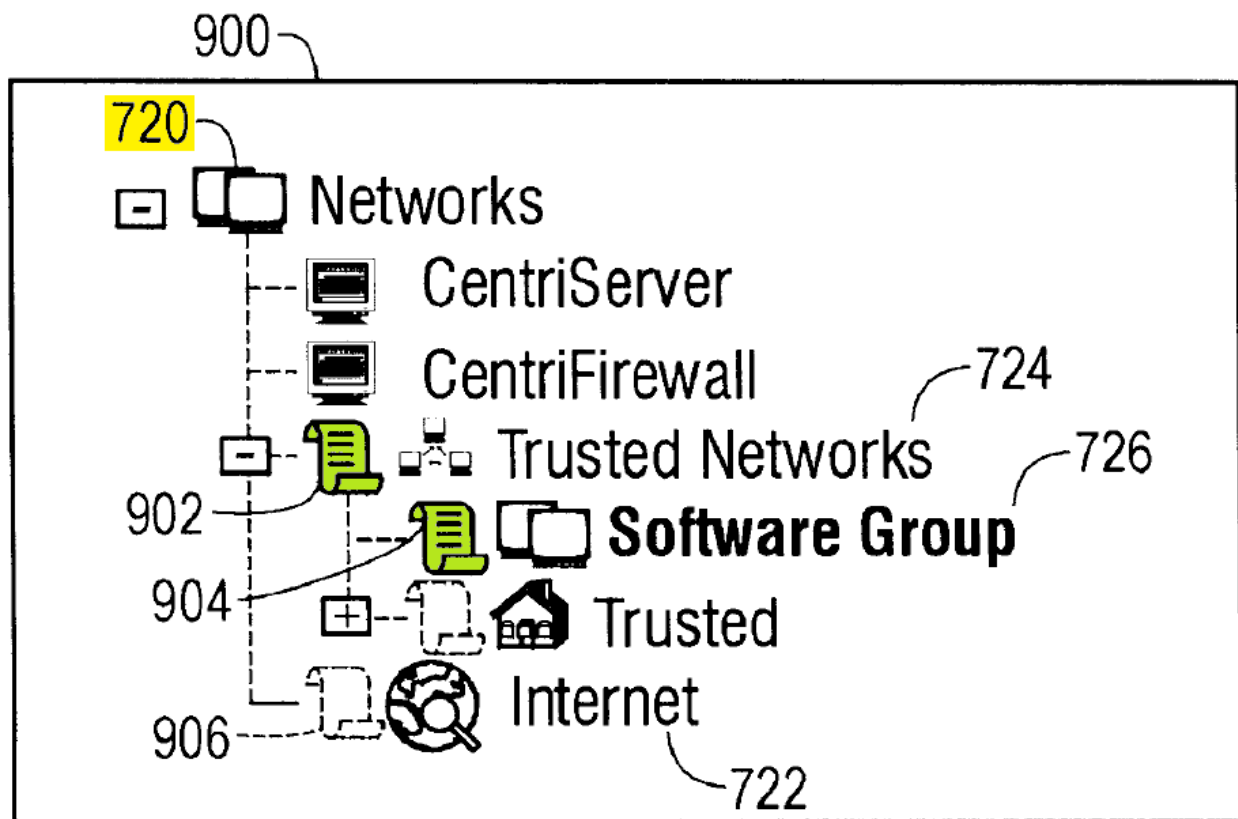


FIG. 9

(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.

123. 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 a metadata to record that security policies have been applied to such resources – the system otherwise may have no way of knowing which security policies applied to which resources.

124. 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.

125. Now having explained how Farino and Richman would be combined with Wiegel, I will now explain the motivation to do so.

126. ***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, Farino, Richman and Wiegel are all analogous references, and I discussed Farino and Richman in the discussion of claim 11[b] above. With respect to Wiegel, it is an analogous reference in the field of systems for creating and managing security policies, and would have been reasonably pertinent to a number of problems facing the inventor. For example, Wiegel would have been reasonably pertinent to the problem of how to design and implement visual rules editor for creating security policies.

127. With respect to 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 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 adapt the Farino-Richman physical security system based on the teachings of Wiegel to provide a straightforward user interface to make it easier to define security policies for the site.

128. I note that Wiegel concerns security policies for network/IT devices such as network devices, routers, and so forth, 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. This is further supported by the fact that the unified access control server in Farino was specifically designed to enforce both physical and network security policies. (Farino, *e.g.*, ¶¶0062 (“Server **200** also enforces all access policies. Further, the unified access control server **200** allows for physical access events and network access events to be monitored, correlated, and tied together to strengthen **both physical and network access control policies.**”), 0066.) The benefits of such unification would have been apparent; for example, if a

user's network access privileges were revoked, the unified system of Farino could also revoke that user's physical access privileges, or vice versa. (Farino, ¶0048 (noting benefits of linking network and physical security processes).) 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.

129. An ordinarily skilled artisan would have perceived no technological obstacle to this combination, and would have had every expectation of success. The ability to create graphical user interfaces such as the one described in Wiegel were basic knowledge of ordinarily skilled artisans, and Wiegel itself provides extensive detail on how to implement his teachings. An ordinarily skilled artisan would have found that the disclosures that I have cited from Farino, Richman and Wiegel provide more than sufficient detail to implement the techniques they describe upon which I rely, without requiring undue experimentation.

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

130. Now turning back to the final clause of claim 11[f], it does not add much that was not already addressed.

131. Under the proposed combination, the “**standardized rule definitions,**”

which are embodied in security policies as discussed above, “**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 security scripts.

132. The “**actionable events,**” as described for claim 11[e], correspond for example to requests to access particular resources or facilities. Under the proposed combination, as explained fully for claim 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 as described in Richman. Under the proposed combination, therefore, the security policy would have been **applied** to the actionable event reflected in the normalized event data. As Farino explains:

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])**

133. Claim 11[g] was largely covered by the discussion of claim 11[f] just provided above. 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,”** as claimed.

134. I understand that in the pending litigation, Petitioner has argued 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.”* I understand that Patent Owner does not agree with these constructions, and believes the phrase **“transformation and routing rules”** should not be separated and

otherwise requires no express construction. In any case, the prior art renders obvious the claimed rules under either view.

135. The security policies under the proposed combination 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 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... via network the frame-/packet-based network.” (Farino, ¶0067; *see also* Farino, ¶0124 (describing return XML message granting access).) The security policies clearly provide **routing** rules because they control data flow between unified access control server 200 and other network-connected systems, *e.g.*, security devices requesting access from the unified access control server. The security policies also provide **transformation** rules because they process and transform/map incoming event data (such as the “access request message” in Farino discussed above) to appropriate outgoing data (*e.g.* “a corresponding access control response message” as disclosed in Farino) sent over the network. (Richman, ¶¶120-0126.) As explained, the unified

access control server processes and transforms the access request message into an appropriate access control return messages in accordance with the policy.

136. A person of ordinary skill in the art would have appreciated that security policies under the proposed combination comprise “**condition-action sequences**,” *i.e.*, logic specifying actions be taken if policy conditions are satisfied. 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 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).”).) This is the basic structure of not only a security policy, but the “IF-THEN” logic in any computer program. In Wiegel, in fact, the security policy is explicitly defined as a 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 (“A complex condition statement evaluates a session request against two or more conditions that are linked using either a nested IF or an OR conjunction. Complex condition statements allow you to evaluate a session request against multiple conditions.”).)

137. 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 requests for access (“**actionable events**”), such as whether or not to permit access. (Farino, ¶¶0097, 0084, 0026, 0092, 0068.) 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.

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.”**

138. 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 (“Computer **151**, and/or other network infrastructure devices and/or DCP 119 are responsible for implementing the unified access control server’s returned instructions (at a facility, for example, by either setting off the alarm or opening the door.”).)

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.”**

139. 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,**” to which I turn below.

140. 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:

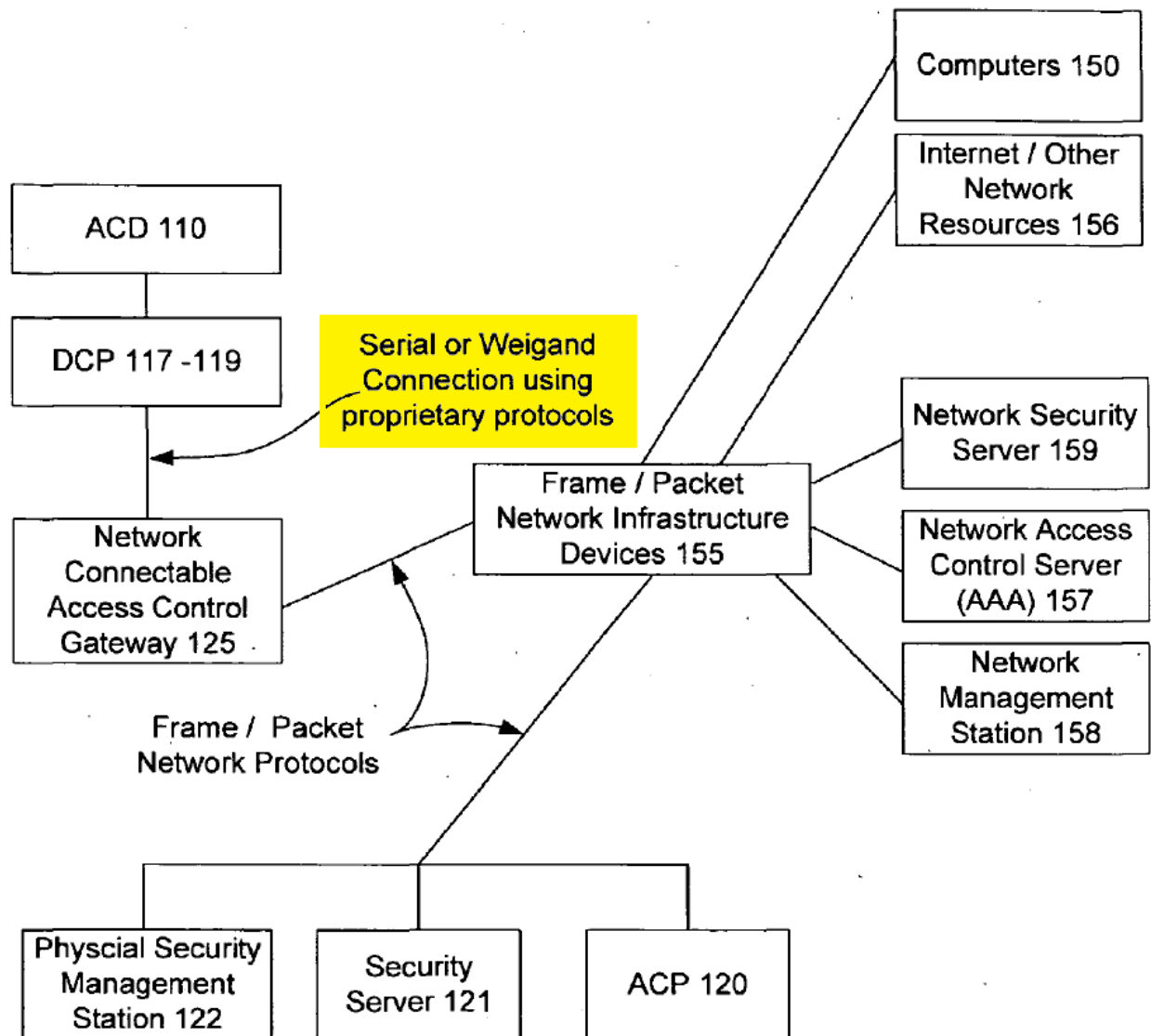


**FIGURE 3
(PRIOR ART)**

(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.) Although figure 3 above includes a “prior art” label, as fully explained for claim 11**[b]**, access control gateway **125** 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).) As explained further below, the native sensor data includes **text data** and **binary data**.

141. The proprietary “Wiegand” format referenced in Farino refers to a well-known technique for sending information from devices such as access control card readers. As explained in **Davis** (EX1009), which I have cited for background information that would have been known to an ordinarily skilled artisan:

The Wiegand standard protocol well known in the art and is described in detail in “Access Control Standard Protocol for the 26-Bit Wiegand Reader Interface,” by the Security Industry Association. The data bits of the transmission from the reader to the panel typically consists of one or more parity bits and numerous data bits, as described in the aforementioned standard. The definition of the data bits are left to the system designer.

(Davis, ¶0003.) It therefore 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 of data transmitted in Wiegand format. For example, at least the “one or more parity bits” in the Wiegand format as discussed above clearly qualify as binary data. Parity bits refer to additional bits that are added to a transmitted message to allow the receiving device to detect if an error occurred that corrupted the data in transmission. But more fundamentally, even without reference to Weigand format, it would have been plainly obvious to an ordinarily skilled artisan that native sensor data format would have included “**binary data,**” as binary data is fundamental to the representation of data in all computer-based systems. (*Microsoft Computer Dictionary* (5th ed. 2002) (EX1010), p.58 (“**binary format.** *n.* Any format that structures data in 8-bit form. Binary format is generally used to represent object code... or data in a transmission stream.”).)

142. 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 combination of Farino and Richman, the original (native) data provided by the sensor (carried for example in the vendor-specific “data bits” of Wiegand format) may have included textual content. 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 (“Physical access control system credentials may be a plastic card encoded with identifying information, a secret code or password entered at a keypad or other biometric information...”); Richman, ¶0120 (“Suppose a tenant wants to access room 5 of a building. The access room is secured with a keypad, which asks for a password and user name.”).) Because the original sensor data collected at the sensor expressly included textual information obtained directly from the user, it would have been plainly obvious that 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 entered textual data matches the stored credentials for the 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.”).) The binary and text data in the native format is “**represented in a proprietary format**,” for the reasons explained above.

143. 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 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 from a computer **151** or a **DCP 119**

connected to access control gateway 225....”).) 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.

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.”**

144. Claim 14 does not add much to what was already discussed for claim 11[g], in which I 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. 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 that executes the security policy that contains the 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.”).)

145. 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 also 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.)

146. 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. Computers generally execute programs in sequential order, *i.e.*, each instruction of a program is carried out in the order in which it is listed in the program.

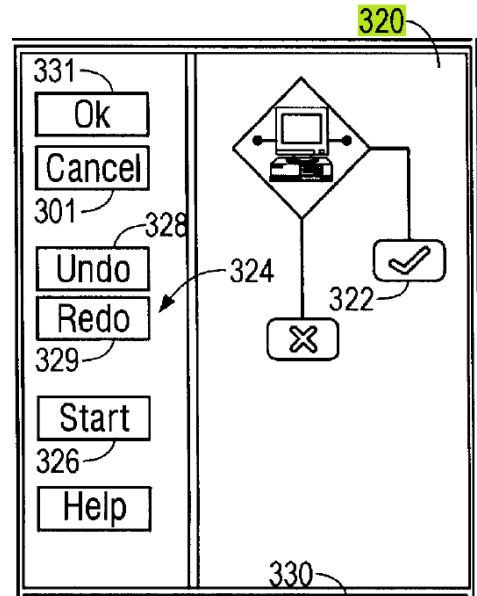
This characteristic also allows programs to nest conditional rules in order to create more complex logic. Wiegel, for example, discloses the ability link multiple rules in a security policy through multiple nested multiple IF-THEN statements for execution in a particular sequence. (Wiegel, 18:1-24.) The ability to sequence such condition-action sequences in this manner was elementary knowledge to ordinarily skilled artisans. 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 logic in the security policy script.

C. Ground 2: Obviousness of Claim 15 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.”**

147. As fully explained this in the discussion of claim 11[f], Figure 3 of

Wiegel depicts policy builder pane 320 (also shown 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 the surrounding user interface features as shown in Figures 3 and 9 that further allow the administrator to apply created security policies to physical security and network devices.

148. 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 fully outlined 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 great detail for claim 11[d].

149. 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 on the screen. Turning back to the example of Figure 9 in Wiegel discussed for claim 11[f], it shows what happens after dragging security policy 319 onto an icon in the network tree:

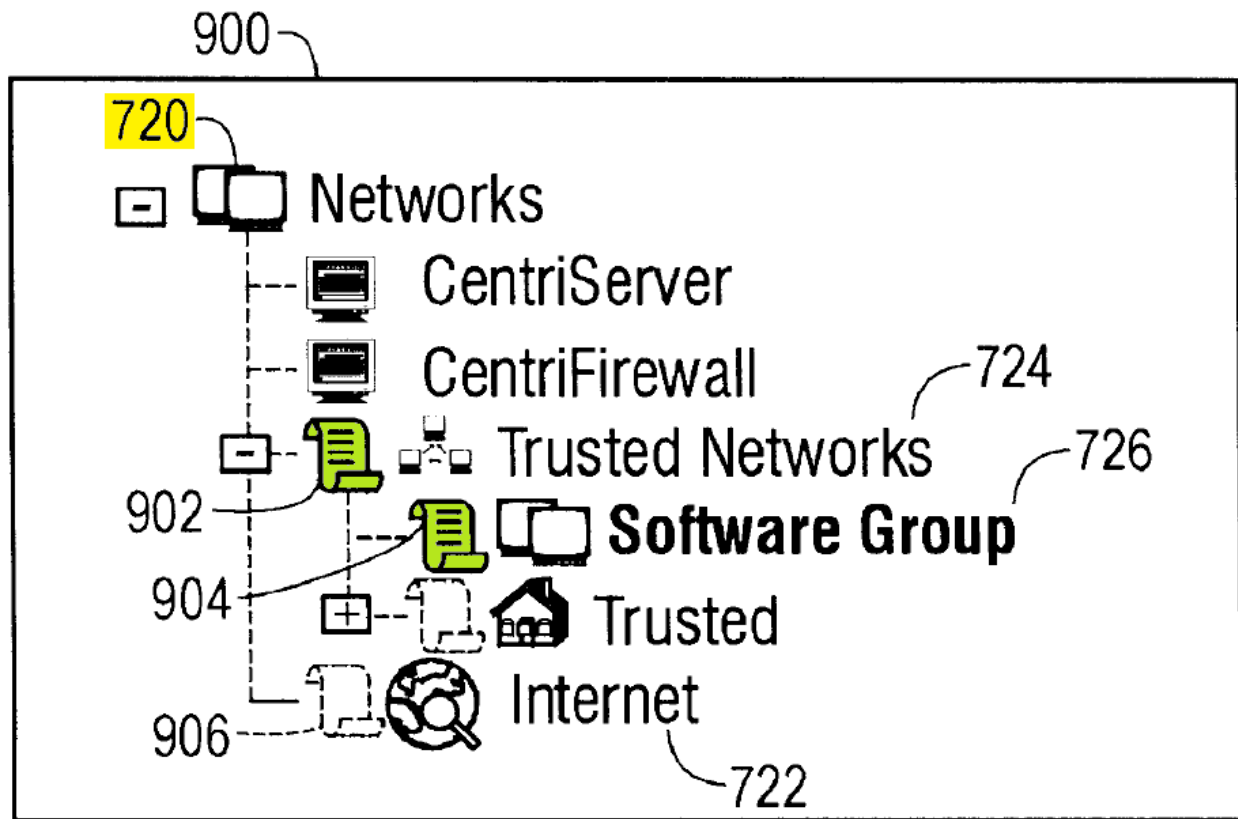


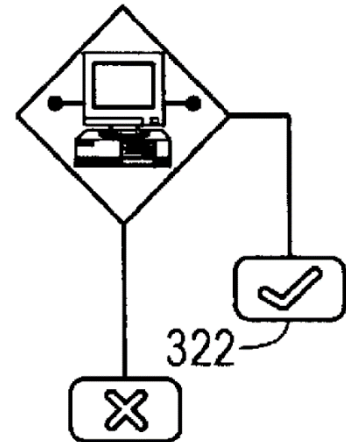
FIG. 9

(Wiegel, Fig. 9 (highlighting added), 27:34-46.) It would have been obvious to an ordinarily skilled artisan 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 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) such a metadata to record how these resources relate to one another within the hierarchy (e.g., which resources fall

are children of other resources in the hierarchical tree), in order to visually depict this relationship to the user. 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 in Wiegel. (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. 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 defined within the software that decides where and how to display on-screen objects. The “**defined rules**” in this case specify how and where to display hierarchically-organized objects and, where applicable, adjacent security policy objects.

150. The claim continues by reciting 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 discloses and renders obvious each aspect of this claim language with the exception of “**a drag and drop graphical user interface component.**” I will thus address that portion separately in connection with the further combination with Moore (EX1006), below.

151. 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 the diamond-shaped “condition symbols” (such as the service symbol shown at right), as well as the “action symbols” (shown as the X and checkmark shown at right). (Wiegel, 16:40-56 (describing condition symbols), 16:57-65 (describing action symbols), Fig. 5 (showing more complex exemplary security policy).)



152. 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:

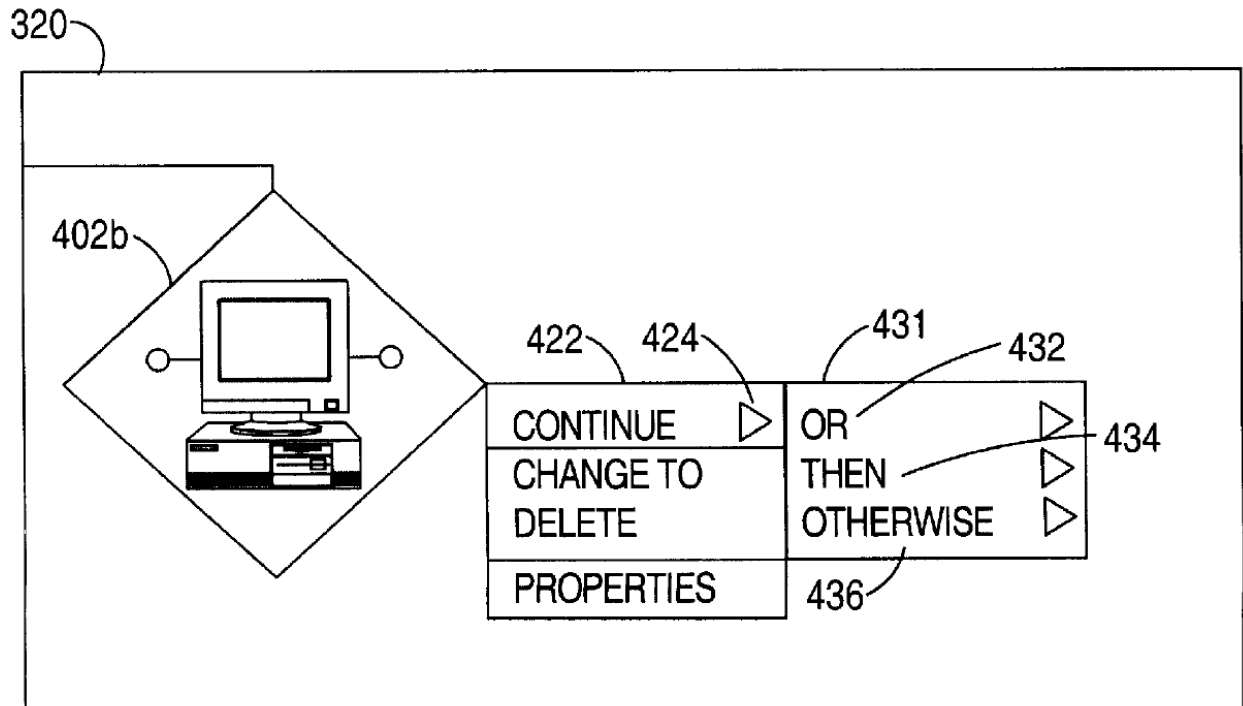


FIG. 4C

(Wiegel, Fig. 4C.) Figure 4C shows a pop-up menu 422 displayed in response to the user right-clicking and thus 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.)

153. 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.)

154. Wiegel thus discloses “**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].” Under the proposed combination, therefore, “**the visual editor... facilitates creation of rules through interconnection of rule objects and definition of rule properties,**” as recited in claim 15.

155. The only aspect of claim 15 not disclosed by the Farino-Richman-Wiegel combination is the further 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 the pop-up menus

through the policy builder pane **320** in Wiegel discussed above. But it would have been obvious, in further view of **Moore** (EX1006), to adapt the user interface to allow these functions to be performed through a drag-and-drop user interface.

156. As would have been known to ordinarily skilled artisans, as well as even the most novice computers users, “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) and “**drag**” it to another location on the screen. The user can then “**drop**” the object by deselecting it (for example by releasing the mouse button), which causes the software to perform an operation. Drag and drop dates back to at least the early Apple Macintosh computers introduced in the mid-1980s, which used drag and drop for common file system operations (*e.g.* moving files between folders, moving files to the trash for deletion, etc.). One of the key benefits of drag and drop is its ease-of-use because it mimics how human beings manipulate physical objects in the real world. It is thus exceptionally easy to learn and use.

157. **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



(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 the popular Apple iTunes program). 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,” “Album,” etc.). I have highlighted an exemplary artist 649 (“Kruder & Dorfmeister”) in yellow, and song (“Encounter”) in green, for illustration.

158. Moore provides an example in which the user assigns the song “Encounter” to the artist “Kruder & Dorfmeister,” using drag and drop:

FIG. 19



(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 item in this case (the song “Encounter”) “does not disappear from the list view; the behavior is more like tagging information to a file.” (Moore, ¶0035.) In other words, the song remains in the item list – the drag and drop operation merely assigned the “Kruder & Dorfmeister” property to that song. Moore thus discloses a simple drag and drop user interface component for assigning properties (e.g., artists) to items (e.g. songs).

159. Although Moore describes his system in the context of organizing digital music, Moore makes clear that the technique is not so limited and can be

applied broadly to any type of computer system in which “data items are organized and queried by way of metadata properties.” (Moore, ¶0010.) The property tree **203** can also be used, for example, to assign properties to “documents, messages, video files, and contacts, with the ProperTree in each case being specifically adapted for the kind of data item that is presented.” (Moore, ¶0022.) In another embodiment, the property tree is a standard control incorporated into an operating system (such as Microsoft Windows), “for application developers to reuse in their own applications.” (Moore, ¶0022.) An ordinarily skilled artisan would have understood that the property tree would have been applicable to any type of data items, including the rule objects in the visual editor under the proposed combination.

160. ***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 described for **Ground 1**, predictably resulting in the security system under the proposed combination in which the policy builder pane interface was adapted to further include “**a drag and drop user interface component that facilitates creation of rules through interconnection of rule objects and definition of rule properties,**” as claimed. The combination would have been straightforward, and an ordinarily skilled artisan would have been motivated to make it, as I will explain below.

161. As noted above, based on the user interface described in Wiegel, the **“interconnection of rule objects and the definition of rule properties”** occurs using pop-up menus 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. To take one example, instead of requiring that the user select a particular condition symbol and navigate through pop-up windows to assign a property to that symbol, as Wiegel discloses, the user could have instead dragged the condition symbol and dropped it onto an appropriate object to make that property assignment. In the example discussed above, the “OR” “THEN” and “OTHERWISE” options in Wiegel for a condition symbol (Fig. 4B) could have been presented as graphical icons onto which a condition symbol could have been dropped.

162. The drag and drop user interface in Moore, moreover, would have been particularly suitable for Wiegel because it allows for hierarchical selections. In the example from Moore discussed above, the user assigned the “**Kruder & Dorfmeister**” artist property to the “**Encounter**” song by dragging the song onto that property. As shown in Figures 18 and 19, that drag and drop operation can cause expansion of hierarchical item lists:

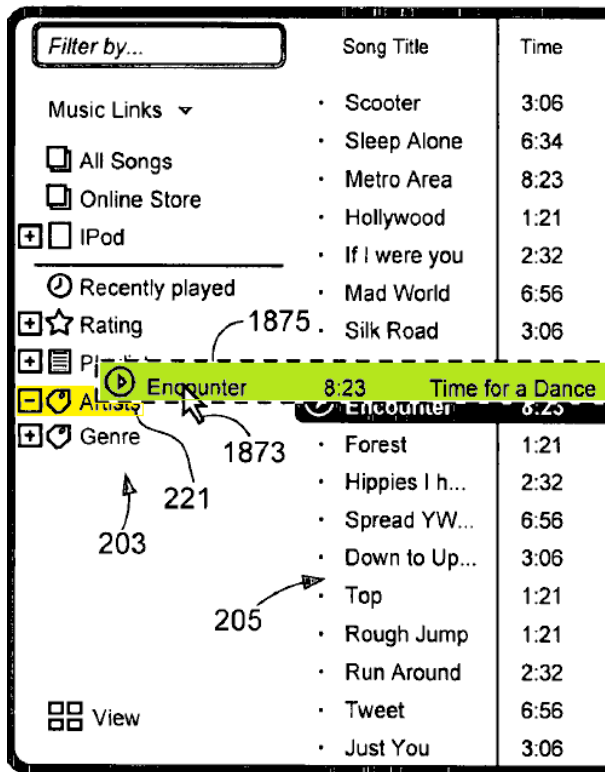


FIG. 18

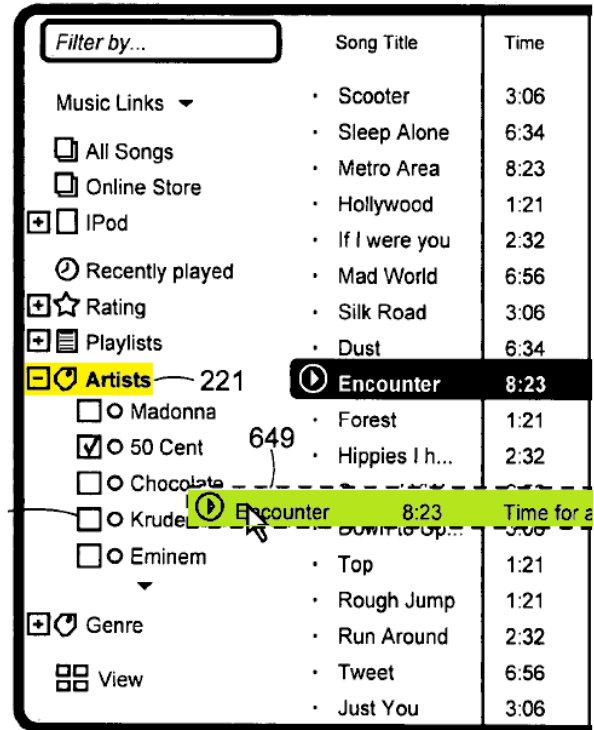


FIG. 19

(Moore, Figs. 18-19 (partial figures, highlighting added).) Figure 18 shows that when the user began the drag and drop operation, the Artists **221** node in the property tree **203** has not yet been expanded. But in Figure 19, as shown, the drag and drop operation caused Artists **221** node in the property tree **203** to expand and reveal the five available artist properties, facilitating the user's selection. (Moore, ¶0035.) Moore explains that the property tree "supports free-form hierarchical properties that are set by the user." (Moore, ¶0041.)

163. This feature makes Moore particularly adaptable to the proposed combination because drag and drop could have provided a straightforward

replacement for the pop-up menus 422 of Wiegel, thus allowing drag condition symbols and drop them onto objects for the “OR” “THEN” and “OTHERWISE” commands in Wiegel, which could have further expanded hierarchically to the extent these commands required additional or nested selections. (Wiegel, Figs. 4C-4G.)

164. Moore is an analogous reference in the field of computer user interfaces, and more specifically, user interfaces for assigning properties to data items. The teachings of Moore would have been reasonably pertinent to multiple problems faced by the inventors, including how to design a user interface to allow the user to easily assign properties to items (such as on-screen visual rule objects). Moore, as explained below, provides a simple and straightforward technique using well-known drag and drop techniques that would have been readily usable by the inventors in implementing the claimed visual editor.

165. 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. As noted above, 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.

166. Drag and drop also has advantages over other techniques such as the pop-up menus used in Wiegel. The pop-up menu **422** in Wiegel requires 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.

167. Drag and drop would also have enabled the visual editor to provide greater consistency in the location of on-screen controls. This is because the pop-up menus **422** in Wiegel appear and expand out from the location of the selected condition symbol. (Wiegel, Fig. 4C.) The pop-up menus **422** will thus pop-up in different places on the screen depending on the location of the selected condition symbol, which could have been jarring or confusing to users. This in turn could have been disruptive to users and/or caused them to lose focus. With drag and drop, in contrast, the available property selections could have been placed in a fixed place on the screen (such as a property tree as in Moore), thus providing a more consistent user interface for the assignment of properties across all condition symbols.

168. An ordinarily skilled artisan would have had at least a reasonable expectation of success. As noted, drag and drop functionality dates back to at least the 1980s, and conventional operating systems such as Microsoft Windows provided

built-in APIs to allow developers to incorporate drag and drop techniques into the user interfaces of their applications. The implementation of drag and drop user interface techniques would have fallen within the knowledge of ordinarily skilled artisans. An ordinarily skilled artisan would have found that the disclosures that I have cited from Moore provide more than sufficient detail to implement the techniques they describe upon which I rely within the proposed combination, without requiring undue experimentation.

VII. NO SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

169. I understand from counsel that the Patent Owner in the underlying district court litigation has not yet identified any evidence with respect to secondary considerations of non-obviousness.

170. To the extent Patent Owner cites any evidence of sales or any praise or any industry recognition of products that Patent Owner asserts to implement the claimed invention, I am not aware of any information demonstrating that any increased sales, commercial success, praise, or any other secondary factor (that Patent Owner may assert) was a result of *the particular features recited in the '704 patent's claims*. Since Patent Owner has not yet identified any evidence of secondary considerations, Patent Owner cannot demonstrate that the limitations of the claimed invention in particular, as opposed to other features of the products at

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

issue, were the factors that caused any increased sales, praise, or any other asserted secondary considerations.

171. Thus, based on my review of the evidence to date, I can summarize my opinions regarding any alleged secondary considerations of non-obviousness relating to the '704 patent, as follows:

172. *No commercial success of the claimed invention.* Patent Owner has not cited any evidence of particular commercial success of products embodying the '704 patent as opposed to products that do not embody the '704 patent. Patent Owner has not cited any evidence that any commercial success of any products is particularly a result of the claimed inventions recited in the '704 patent's claims and not due to any other facts.

173. *No long-felt but unsolved need.* Patent Owner has not cited any evidence of any long-felt need that remained unsolved in the prior art before the '704 patent. To the contrary, as discussed above, the prior art solved the problems that the '704 patent purported to address.

174. *No failure of others.* Patent Owner has not cited any evidence of anyone who tried, but failed, to solve the problems addressed by the '704 patent. As shown by my analysis above, there existed prior art references that successfully disclosed and rendered obvious the subject matter claimed by the '704 patent.

175. *No copying of the claimed invention.* Patent Owner has not cited any evidence that any other party (including Facebook or third parties) ever copied from the '704 patent and its claimed invention.

176. *No unexpected results of the claimed invention.* Patent Owner has not cited any evidence of unexpected results achieved by the '704 patent's claimed invention. To the contrary, the prior art disclosed the predictable, expected results that show why the '044 patent's claims are obvious as discussed in my Declaration.

177. *No praise for the claimed invention.* Patent Owner has not cited any evidence of praise for the claimed invention recited in the '704 patent.

178. *No surprise or skepticism at the claimed invention.* Patent Owner has not cited any evidence that observers were surprised by, or skeptical of, the claimed invention recited in the '704 patent.

179. *No departure from the wisdom of the prior art.* Patent Owner has not cited any evidence that the claimed inventions of the '704 patent departed from the wisdom of the prior art. The '704 patent claims subject matter that was already present in the prior art, including in the references discussed in my Declaration.

180. Moreover, with respect to the considerations discussed above, I also refer to and incorporate my opinions stated throughout this Declaration, including my analysis showing that the '704 patent is directed to techniques known in the prior

Declaration of Vijay K. Madiseti, Ph.D. in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 8,234,704 B2

art and does not provide any inventive technology.

181. To the extent Patent Owner at a later date cites or provides any other evidence regarding secondary considerations, including any expert opinions, I reserve the right to supplement my analysis and opinions to comment on it.

VIII. CONCLUSION

182. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the '704 patent claims at issue and/or offer testimony in support of this Declaration.

183. I hereby declare that all statements made herein of my own knowledge are true and that all statements are made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Dated: December 16, 2022

Respectfully submitted,



Vijay K. Madisetti, Ph.D.
Johns Creek, Georgia

EXHIBIT A

Dr. Vijay K. Madisetti
Fellow, IEEE
vkm@madisetti.com
Cell: 770-527-0177
Address:
56 Creekside Park Drive
Johns Creek, GA 30022

Employment:

- 1989-present: Full Professor of Electrical & Computer Engineering (Georgia Tech, Atlanta, GA 30332).

Areas of Technical Interest – Embedded Systems, Wireless & Mobile Communications, Computer Engineering, Circuit Design (Analog/Digital), Biomedical Instrumentation Hardware/Software, Software Engineering, Digital Signal Processing, Sensors, Wireline & Wireless Computer Networks, Software Systems, Control Systems, Chip Design, Cloud Computing.

Education: PhD (EECS, University of California at Berkeley, 1989), B.Tech (Hons) in Electronics and Electrical Communications Engineering (Indian Institute of Technology, Kharagpur, 1984).

Startup Companies:

Director, VP Technologies, Inc. (1995-): A startup commercialized through Georgia Tech's Advanced Technology Development Corporation (ATDC) focusing on digital software and hardware design services for military market. <http://www.vptinc.net>

Director, Soft Networks, LLC (2001-2007): A startup commercialized through Georgia Tech support focusing on software development tools and compilers for Cellular/WiFi/VOIP/telecommunication products. <http://www.soft-networks.com>

Director, Elastic Video Inc. (2007- 2009): A startup commercialized through Georgia Tech's VentureLab (<http://venturelab.gatech.edu>) development image and video processing software for wireless & IP networking.

Litigation Experience (2015-2021) With Testimony

(Note: There may be multiple cases between the parties, e.g., District Court v. ITC, US versus Foreign Cases)

Chrimar v. Adtran, Alcatel, et al.

Technology: Power over Ethernet (2015-2017)

Submitted reports & Deposition & Trial

Chamberlain v. Ryobi/TTI

Case No: 1:16-cv-06097 (ND Illinois)

Expert for Ryobi

Technology: Wireless/IoT/Barrier Movement (2016 – 2019)

Submitted reports & deposition & trial testimony

IOEngine v. IMC/Imation

Case No: cv-14-1572-GMS (US Delaware)

Expert for IMC/Imation

Technology: Networked Storage Device (2016-2017)

Submitted reports & deposition & trial testimony

Huawei v. Samsung

Case No: 3:16-cv-2787-WHO (ND Cal)

Expert for Samsung

Technology: 4G/LTE Random Access Protocols (2016-2019)

Submitted reports & deposition

Hitachi Maxell v. ZTE/Huawei

Case No: 5:16-cv-00178-RWS (ED Texas)

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted reports & deposition (2017 – 2018)

Hitachi Maxell v. Apple

Case No: 5:19-cv-00036-RWS

Expert for Hitachi Maxell

Technology: Digital Cameras

Submitted Expert Reports (2020-)

Qualcomm v. Apple

Case No: 17-ccv-0108-GPC-MDD (SD Cal) Also, Related ITC/FTC Matters

Expert for Qualcomm

Technology: 4G/Wireless Communications/Smartphones (2017-2019)

Submitted Reports and Deposition

Qualcomm v. Apple

Case No: 3:17-cv-01375-DMS-MDD (SD Cal)

Expert for Qualcomm
Technology: 4G/Wireless Communications/Smartphones (2017-2019)
Submitted Reports and Deposition

Optis v. Huawei

Case No: 2:17-cv-123 (E.D. Texas)
Expert for Optis Wireless
Technology: 4G/Video (2017-2019)
Submitted reports & deposition

Beckman Coulter v. Sysmex

Case No: 1:17-cv-24049-DPG (ND Illinois)
Expert for Sysmex
Technology: Medical Instrumentation Automation (2017-present)
Testifying Expert

TQ Delta v. Xyvel/Adtran

Expert for TQ Delta
Technology: DSL Technologies (2018-present)
Testifying Expert

3GL/KPN v. LG, Blackberry, HTC

Expert for 3GL/KPN
Technology: 4G/LTE Protocols (2018-present)
Reports and Deposition / Testifying Expert

Rovi v. Comcast

ITC No. 337-TA-1103 (ITC)
Expert for Rovi
Technology: Digital Video & interactive GUI (Jan 2018-May 2018)
Reports and Deposition

Cirba/Densify v. VMWare

Case No: 1:19-cv-00742-LPS
Expert for Cirba/Densify
Technology: Virtualization (2019-present)
Reports Deposition/PI Hearing / Testifying Expert

Power Integrations v. On Semiconductor

Case No: 17-cv-03189-BLF
Expert for On Semiconductor
Technology: Power Electronics (2018-2019)
Reports & Deposition/Testifying Expert.

Pathway v. Lumens, IPEVO, Aver JDA

Case No: ITC 337-TA-1045

Expert for Lumens JDA

Technology: Document Cameras (2016-2019)

Reports & Deposition (for IPR/ITC, no trial)

Wilan v. Apple

Case No: 3:14-cv-2235

Expert for WiLan

Technology: Voice over LTE/LTE Protocols (2017-2020)

Reports and Testified through Deposition/Trials

St. Lawrence Comm. v. Amazon

Case No. 2:19-CV-00027-JRG

Expert for Amazon

Technology: Multimedia Codecs

Submitted Expert Reports/Deposition (2020-2020)

PanOptis v. Apple

Case No. 2:19-CV-00066-JRG

Expert for PanOptis

Technology: LTE

Submitted Expert Reports/Deposition/Trial (2019-2020)

GAS v. Sprint Communications

Case No. 2:20-cv-00007-RWS

Expert for General Access (GAS)

Technology: LTE

Submitted Expert Reports/Deposition (9/21-)

Additional matters include declarations supporting IPRs at the PTAB for Google (US Patent 8,601,154), On Semiconductor (US Patent 6,212,079), Ubisoft (US Patent 5,490,216), Lumens JDA (2017-2019), Broadcom (WiFi), Sony (US Patent 6,101,534), Kia, (US Patent 5,530,431), Qualcomm, Ericsson, Daimler Benz, VW, Estech, Amazon, Ring, Digital Ally, On Semiconductor, United Patents, Lenovo, BMC Software, BMW, Daimler and Dolby.

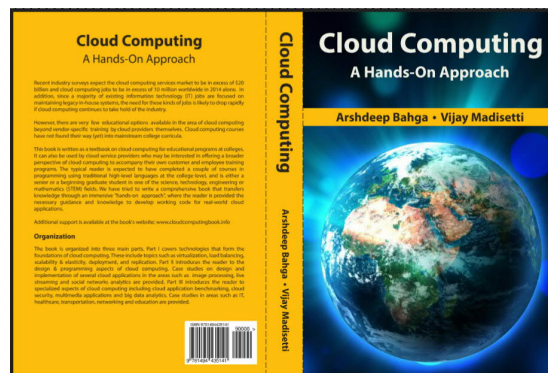
Earned Degrees

1. B. Tech (Hons), Electronics & Electrical Comm. Engineering
Indian Institute of Technology (IIT), Kharagpur, India
1984.

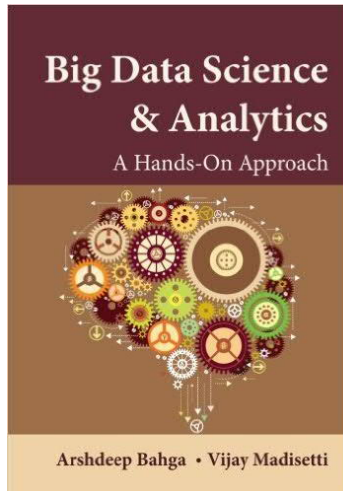
2. Ph.D., Electrical Engineering & Computer Sciences (EECS)
University of California (UC), Berkeley. CA
1989.

Books

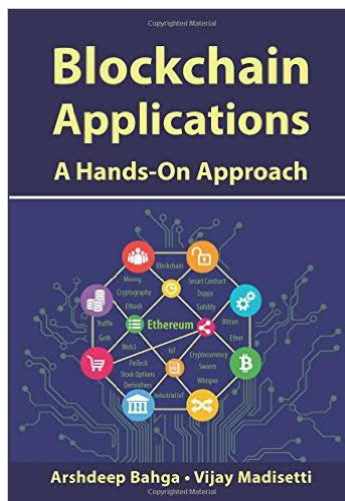
1. VLSI Digital Signal Processors
Madiseti, V.K.;
Boston: MA, IEEE Press: Butterworth Heinemann, 1995, 525 pp.
2. Quick-Turnaround ASIC Design in VHDL
Romdhane, M., Madiseti, V.K., Hines, J.
Boston: MA, Kluwer Academic Press, 1996, 190 pp.
3. The Digital Signal Processing Handbook (First Edition)
Madiseti, V. K., Williams, D. (Editors)
CRC Press, Boca Raton, Fla, 1998, 2500 pp.
4. VHDL: Electronics Systems Design Methodologies.
Madiseti, V. K. (Editor)
Boston: MA, IEEE Standards Press, 2000, ISBN 0-7381-1878-8.
5. Platform-Centric Approach to System-on-Chip (SoC) Design.
Madiseti, V. K., Arpikanondt, A.
Springer, Boston: MA, Springer, 2004, 280 pp.
6. The Digital Signal Processing Handbook – Second Edition.
Madiseti, V. K. (2009)
CRC Press, Boca Raton, Fla.
7. Cloud Computing: A Hands-On Approach
A Bahga, V. Madiseti (2013)
Amazon CreateSpace Publishing, 2013, 454 pp.
8. Internet of Things: A Hands-On Approach
A Bahga, V. Madiseti (2014)
Amazon CreateSpace Publishing, 2014, 450 pp.



9. Big Data Science & Analytics: A Hands-On Approach
A Bahga, V. Madiseti (2016)
Amazon CreateSpace Publishing, 2016, 542 pp.



10. Blockchain Applications: A Hands-On Approach
A Bahga, V. Madiseti (2017)
Amazon CreateSpace Publishing, 2017, 380 pp.



Edited Books & Collection of Papers

1. Advances in Parallel & Distributed Simulation
Madiseti, V.K.; Nicol, D., Fujimoto, R. (Editors)
San Diego, CA: SCS Press, 1991, 200 pp.
2. Modeling, Analysis, Simulation of Computer & Telecommunications Systems
Madiseti, V., Gelenbe, E., Walrand, J. W. (Editors)
Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.

3. Modeling & Simulations on Microcomputers
Madisetti, V.K. (Editor)
San Diego, CA: SCS Press, 138 pp. 1990.

Editorship of Journals & Transactions

1. IEEE Design & Test of Computers
Special Issue: Reengineering Digital Systems
April – June 1999 (Vol 16, No 2)
Madisetti, V.K (Editor)
Los Alamitos: CA, IEEE Computer Society Press, 1999.
2. IEEE Design & Test of Computers
Special Issue: Rapid Prototyping of Digital Systems
Fall 1996 (Vol 13, No 3)
Madisetti, V., Richards, M. (Editors)
Los Alamitos: CA, IEEE Computer Society Press, 1994, 425 pp.
3. IEEE Transactions on Circuits & Systems II
Associate Editor: 1993-1995.
4. International Journal in Computer Simulation
Associate Editor: 1990-1993
5. International Journal in VLSI Signal Processing
Editorial Board: 1995 - Present

Recent Issued US Patents

- [1] [10,503,927](#) [Method and system for securing cloud storage and databases from insider threats and optimizing performance](#), Issued Dec 10, 2019
- [2]. [10,460,283](#) [Smart contract optimization for multiparty service or product ordering system](#), Issued Oct 29, 2019
- [3]. [10,459,946](#) [Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing](#), Issued Oct 29, 2019

- [4]. [10,402,589](#) Method and system for securing cloud storage and databases from insider threats and optimizing performance, Issued Sep 3, 2019
- [5]. [10,394,845](#) Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, Issued Aug 27, 2019
- [6]. [10,289,631](#) Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, Issued May 24, 2019
- [7]. [10,255,342](#) Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing, Issued April 9, 2019
- [8]. [10,243,743](#) Tokens or crypto currency using smart contracts and blockchains, Issued March 26, 2019
- [9]. [10,204,339](#) Method and system for blockchain-based combined identity, ownership, integrity and custody management, Issued February 12, 2019
- [10]. [10,204,148](#) Method and system for tuning blockchain scalability, decentralization, and security for fast and low-cost payment and transaction processing, Issued Feb 12, 2019
- [11]. [10,121,143](#) Method and system for blockchain-based combined identity, ownership, integrity and custody management, Issued Nov 6, 2018
- [12]. [10,102,526](#) Method and system for blockchain-based combined identity, ownership, integrity and custody management, October 16, 2018
- [13]. [10,102,265](#) Method and system for tuning blockchain scalability for fast and low-cost payment and transaction processing, October 16, 2018
- [14]. [9,935,772](#) Methods and systems for operating secure digital management aware applications, April 3, 2018
- [15]. [9,769,213](#) Method and system for secure digital object management, Issued Sep 19, 2017

Refereed Journal Publications

1. Trends in the Electronic Control of Mine Hoists
Madiseti, V. and Ramlu, M.,
IEEE Transactions on Industry Applications, Vol IA-22, No. 6,
November/December 1986. Pages 1105-1112
2. Multilevel range/NEXT performance in digital subscriber loops
Brand, G.; Madiseti, V.; Messerschmitt, D.G.;
Communications, Speech and Vision, IEE Proceedings I [see also IEE

- Proceedings-Communications] ,Volume: 136 , Issue: 2 , April 1989
Pages:169 – 174
3. Seismic migration algorithms on parallel computers
Madiseti, V.K.; Messerschmitt, D.G.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 39 , Issue: 7 , July 1991
Pages:1642 – 1654
 4. Asynchronous algorithms for the parallel simulation of event-driven
dynamical systems
Madiseti, V.K.; Walrand, J.C.; Messerschmitt, D.G.:
ACM Transactions on Modeling and Computer Simulation, v 1, n 3, July 1991,
Pages: 244-74
 5. Synchronization mechanisms for distributed event-driven
computation
Madiseti, V.K.; Hardaker, D.:
ACM Transactions on Modeling and Computer Simulation, v 2, n 1, Jan. 1992,
Pages: 12-51
 6. Efficient VLSI Architectures for the Arithmetic Fourier Transform
(AFT)
Kelley, B.T.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 1 , January
1993
Pages:365-378
 7. The fast discrete Radon transform. I. Theory
Kelley, B.T.; Madiseti, V.K.;
Image Processing, IEEE Transactions on ,Volume: 2 , Issue: 3 , July 1993
Pages:382 – 400
 8. The Georgia Tech digital signal multiprocessor
Barnwell, T.P., III; Madiseti, V.K.; McGrath, S.J.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 41 , Issue: 7 , July 1993
Pages:2471 – 2487
 9. The MIMDIX Environment for Parallel Simulation
Madiseti, V.K.; Hardaker, D.; Fujimoto, R.M.:
Journal of Parallel and Distributed Computing, v18, no. 4, August 1993,
Pages: 473-83.

10. LMSGEN: a prototyping environment for programmable adaptive digital filters in VLSI
Romdhane, M.S.B.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:33 – 42
11. Fixed-point co-design in DSP
Egolf, T.W.; Famorzadeh, S.; Madiseti, V.K.;
Chapter in VLSI Signal Processing, VII, 1994.,
Pages:113 - 126
12. A fast spotlight-mode synthetic aperture radar imaging system
Madiseti, V.K.;
Communications, IEEE Transactions on ,Volume: 42 , Issue: 234 , February-April 1994
Pages:873 – 876
13. Rapid prototyping on the Georgia Tech digital signal multiprocessor
Curtis, B.A.; Madiseti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 3 , March 1994
Pages:649 – 662
14. Low-power signaling in asymmetric noisy channels via spectral shaping
Sipitca, M.; Madiseti, V.K.;
Signal Processing Letters, IEEE, Volume: 1 , Issue: 8 , Aug 1994
Pages:117 – 118
15. A quantitative methodology for rapid prototyping and high-level synthesis of signal processing algorithms
Madiseti, V.K.; Curtis, B.A.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 42 , Issue: 11 , Nov. 1994
Pages:3188 – 3208
16. Computer Simulation of Application-Specific Signal Processing Systems
Casinovi, G.; Madiseti, V.K.;
International Journal in Computer Simulation, Vol. 4, No. 4, Nov 1994.
17. System partitioning of MCMs for low power
Khan, S.A.; Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 12 , Issue: 1 , Spring 1995
Pages:41 – 52

18. Error correcting run-length limited codes for magnetic recording
Jaejin Lee; Madiseti, V.K.;
Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 6 , Nov. 1995
Pages:3084 – 3086
19. Virtual prototyping of embedded microcontroller-based DSP systems
Madiseti, V.K.; Egolf, T.W.;
Micro, IEEE ,Volume: 15 , Issue: 5 , Oct. 1995
Pages:9 – 21
20. Constrained multitrack RLL codes for the storage channel
Lee, J.; Madiseti, V.K.;
Magnetics, IEEE Transactions on ,Volume: 31 , Issue: 3 , May 1995
Pages:2355 – 2364
21. Rapid digital system prototyping: current practice, future challenges
Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:12 – 22
22. Conceptual prototyping of scalable embedded DSP systems
Dung, L.-R.; Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:54 – 65
23. Advances in rapid prototyping of digital systems
Madiseti, V.K.; Richards, M.A.;
Design & Test of Computers, IEEE ,Volume: 13 , Issue: 3 , Fall 1996
Pages:9
24. Combined modulation and error correction codes for storage channels
Jaejin Lee; Madiseti, V.K.;
Magnetics, IEEE Transactions on ,Volume: 32 , Issue: 2 , March 1996
Pages:509 – 514
25. Model-based architectural design and verification of scalable embedded DSP systems-a RASSP approach
Dung, L.-R.; Madiseti, V.K.; Hines, J.W.;
Chapter in VLSI Signal Processing, IX, 1996.
Pages:147 – 156

26. Low-power digital filter implementations using ternary coefficients
Hezar, R.; Madisetti, V.K.;
Chapter in VLSI Signal Processing, IX, 1996.,
Pages:179 – 188
27. All-digital oversampled front-end sensors
Romdhane, M.S.B.; Madisetti, V.K.;
Signal Processing Letters, IEEE, Volume: 3 , Issue: 2 , Feb. 1996
Pages:38 – 39
28. Modeling COTS components in VHDL
Calhoun, S., Reese, R; Egolf, T., Madisetti, V.K.;
Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996
Pages: 24 – 31
29. VHDL-Based Rapid Systems Prototyping
Egolf, T.; Madisetti, V.K.;
Journal of VLSI Signal Processing, Volume: 14 , Issue: 2 , Nov 1996
Pages: 40-52
30. Interface design for core-based systems
Madisetti, V.K.; Lan Shen;
Design & Test of Computers, IEEE ,Volume: 14 , Issue: 4 , Oct.-Dec. 1997
Pages:42 - 51
31. Incorporating cost modeling in embedded-system design
Debardelaben, J.A.; Madisetti, V.K.; Gadiant, A.J.;
Design & Test of Computers, IEEE ,Volume: 14 , Issue: 3 , July-Sept. 1997
Pages:24 – 35
32. On homomorphic deconvolution of bandpass signals
Marenco, A.L.; Madisetti, V.K.;
Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and
Signal Processing, IEEE Transactions on] ,Volume: 45 , Issue: 10 , Oct. 1997
Pages:2499 – 2514
33. A case study in the development of multi-media educational material:
the VHDL interactive tutorial
Gadiant, A.J.; Stinson, J.A., Jr.; Taylor, T.C.; Aylor, J.H.; Klenke, R.H.;
Salinas, M.H.; Madisetti, V.K.; Egolf, T.; Famorzadeh, S.; Karns, L.N.; Carter,
H.W.;
Education, IEEE Transactions on ,Volume: 40 , Issue: 4 , Nov. 1997
Pages:17 pp.

34. Adaptive mobility management in wireless networks
Jeongwook Kim; Madiseti, V.K.;
Electronics Letters ,Volume: 34 , Issue: 15 , 23 July 1998
Pages:1453 – 1455
35. Efficient implementation of two-band PR-QMF filterbanks
Hezar, R.; Madiseti, V.K.;
Signal Processing Letters, IEEE ,Volume: 5 , Issue: 4 , April 1998
Pages:92 – 94
36. On fast algorithms for computing the inverse modified discrete cosine transform
Yun-Hui Fan; Madiseti, V.K.; Mersereau, R.M.;
Signal Processing Letters, IEEE ,Volume: 6 , Issue: 3 , March 1999
Pages:61 – 64
37. System on chip or system on package?
Tummala, R.R.; Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:48 – 56
38. Reengineering legacy embedded systems
Madiseti, V.K.; Jung, Y.-K.; Khan, M.H.; Kim, J.; Finnessy, T.;
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:38 – 47
39. Reengineering digital systems
Madiseti, V.K.;
Design & Test of Computers, IEEE ,Volume: 16 , Issue: 2 , April-June 1999
Pages:15 – 16
40. Parameter optimization of robust low-bit-rate video coders
Sangyoun Lee; Madiseti, V.K.;
Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 9
Issue: 6 , Sept. 1999
Pages:849 – 855
41. Closed-form for infinite sum in bandlimited CDMA
Jatunov, L.A.; Madiseti, V.K.;
Communications Letters, IEEE ,Volume: 8 , Issue: 3 , March 2004
Pages:138 – 140
42. A new protocol to enhance path reliability and load balancing in mobile ad hoc networks
Argyriou, A.; Madiseti, V.K.;

Journal of Ad Hoc Networks, Elsevier Press, 2004

43. Closed-form analysis of CDMA systems using Nyquist pulse
Jatunov, L.A.; Madiseti, V. K.;
Communications Letters, IEEE (Under Revision), 2005.
44. Systematic Design of End-to-End Wireless Mobility Management
Protocols,
Argyriou, A.; Madiseti, V. K.;
ACM/Springer Wireless Networks (WINET), Accepted 2005.
45. A Novel End-to-End Approach for Video Streaming Over the Internet,
Argyriou, A.; Madiseti, V. K.;
Kluwer Telecommunications Systems, Vol. 28, No. 2, Pages 133-150, Jan
2005. Special Issue on Multimedia Streaming.
46. An Analytical Framework of RD Optimized Video Streaming with TCP,
Argyriou, A.; Madiseti, V. K.;
IEEE Transactions on Multimedia, Submitted for review in March 2005.
47. Modeling the Effect of Handoffs on Transport Protocol Performance,
Argyriou, A.; Madiseti, V. K.;
IEEE Transactions on Mobile Computing, Submitted for review in March 2005
48. Throughput Models for Transport Protocols with CBR and VBR Traffic
Workloads",
Argyriou, A.; Madiseti, V. K.;
ACM Transactions on Multimedia Computing, Communications & Applications,
Submitted for review in April 2005.
49. "Electronic System, Platform & Package Codesign",
Madiseti, V. K.
IEEE Design & Test of Computers, Volume 23, Issue 3, June 2006. pages
220-233.
50. "The Design of an End-to-End Handoff Management Protocol",
A. Argyriou, Madiseti, V. K.
Wireless Networks, Springer, May 2006.
51. "A Soft-Handoff Transport Protocol for Media Flows in Heterogeneous
Mobile Networks ",
A. Argyriou, Madiseti, V. K.
Computer Networks, Vol 50, Issue 11, Pages 1860-1871, August 2006.

52. "Computationally Efficient SNR Estimation for Bandlimited WCDMA Systems"
L. Jatunov, Madiseti, V. K.
IEEE Transactions on Wireless Communications, Volume 5, Issue 13,
December 2006, Pages 3480-3491.
53. "Space-Time Codes for Wireless & Mobile Applications",
M. Sinnokrot, Madiseti, V.K.
DSP Handbook, Second Edition, 2009 (to be published)
- 54.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems", IEEE Computer, vol. 46, no. 11, Nov 2013, pp 76-83, 2013
- 55.A. Bahga, V. Madiseti, "Cloud-Based Information Integration & Informatics Framework for Healthcare Applications", IEEE Computer, February 2015.
- 56.A. Bahga, V. Madiseti, "A Cloud-based Approach for Interoperable EHRs", IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 57.A. Bahga, V. Madiseti, "Cloud-Based Information Technology Framework for Data Driven Intelligent Transportation Systems", Journal of Transportation Technologies, vol.3 no.2, April 2013
- 58.A. Bahga, V. Madiseti, "Performance Evaluation Approach for Multi-tier Cloud Applications", Journal of Software Engineering and Applications, vol. 6, no. 2, pp. 74-83, Mar 2013.
- 59.Yusuf, A., V. Madiseti, "Configuration for Predicting Travel Time Using Wavelet Packets and Support Vector Regression", Journal of Transportation Technologies, vol 3, no. 3, June 2013.
- 60.A. Bahga, V. Madiseti, "Analyzing Massive Machine Maintenance Data in a Computing Cloud", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 10, pp. 1831 - 1843, 2012.
- 61.N. Radia. Y. Zhang, M. Tatimapula, V. Madiseti, "Next Generation Applications on Cellular Networks: Trends, Challenges, and Solutions," Proceedings of the IEEE, Vol 100, Issue 4, pp. 841-854, 2012.
- 62.A. Bahga, V. Madiseti, "Rapid Prototyping of Advanced Cloud-Based Systems", IEEE Computer, vol. 46, no. 11, Nov 2013, pp 76-83, 2013

- 63.A. Bahga, V. Madiseti, "A Cloud-based Approach for Interoperable EHRs", IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 5, Sep 2013, pp. 894-906, 2013
- 64.A. Bahga, V. Madiseti, "Cloud-Based Information Integration & Informatics Framework for Healthcare Applications", IEEE Computer, February 2015.

Peer Reviewed Conference Publications

1. Dynamically-reduced complexity implementation of echo cancelers
Madiseti, V.; Messerschmitt, D.; Nordstrom, N.;
Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86. ,Volume: 11 , Apr 1986
2. Seismic migration algorithms using the FFT approach on the NCUBE multiprocessor
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on , 11-14 April 1988
3. Seismic migration algorithms on multiprocessors
Madiseti, V.K.; Messerschmitt, D.G.;
Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on , 11-14 April 1988
Pages:2124 - 2127 vol.4
4. WOLF: A rollback algorithm for optimistic distributed simulation systems
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Conference Proceedings, 1988 Winter , December 12-14, 1988
Pages:296 – 305
5. Efficient distributed simulation
Madiseti, V.; Walrand, J.; Messerschmitt, D.;
Simulation Symposium, 1989. The 22nd Annual , March 28-31, 1989
Pages:5 - 6
6. High speed migration of multidimensional seismic data
Kelley, B.; Madiseti, V.;

- Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991
International Conference on , 14-17 April 1991
Pages:1117 - 1120 vol.2
7. Performance of a fast analog VLSI implementation of the DFT
Buchanan, B.; Madiseti, V.; Brooke, M.;
Circuits and Systems, 1992., Proceedings of the 35th Midwest Symposium on
, 9-12 Aug. 1992
Pages:1353 - 1356 vol.2
 8. Task scheduling in the Georgia Tech digital signal multiprocessor
Curtis, B.A.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on , Volume: 5 , 23-26 March 1992
Pages:589 - 592 vol.5
 9. The fast discrete Radon transform
Kelley, B.T.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE
International Conference on , Volume: 3 , 23-26 March 1992
Pages:409 - 412 vol.3
 10. Yield-based system partitioning strategies for MCM and ASEM design
Khan, S.; Madiseti, V.;
Multi-Chip Module Conference, 1994. MCMC-94, Proceedings., 1994 IEEE, 15-17 March 1994
Pages:144 - 149
 11. Multitrack RLL codes for the storage channel with immunity to intertrack interference
Lee, J.; Madiseti, V.K.;
Global Telecommunications Conference, 1994. GLOBECOM '94.
'Communications: The Global Bridge', IEEE, Volume: 3 , 28 Nov.-2 Dec. 1994
Pages:1477 - 1481 vol.3
 12. A parallel mapping of backpropagation algorithm for mesh signal processor
Khan, S.A.; Madiseti, V.K.;
Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop , 31 Aug.-2 Sept. 1995
Pages:561 - 570
 13. Virtual prototyping of embedded DSP systems
Madiseti, V.K.; Egolf, T.; Famorzadeh, S.; Dung, L.-R.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995

- International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2711 - 2714 vol.4
14. Assessing and improving current practice in the design of application-specific signal processors
Shaw, G.A.; Anderson, J.C.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2707 - 2710 vol.4
15. Introduction to ARPA's RASSP initiative and education/facilitation program
Corley, J.H.; Madiseti, V.K.; Richards, M.A.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 4 , 9-12 May 1995
Pages:2695 - 2698 vol.4
16. DSP design education at Georgia Tech
Madiseti, V.K.; McClellan, J.H.; Barnwell, T.P., III;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995
International Conference on ,Volume: 5 , 9-12 May 1995
Pages:2869 - 2872 vol.5
17. Rapid prototyping of DSP systems via system interface module generation
Famorzadeh, S.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1256 - 1259 vol. 2
18. Rapid prototyping of DSP chip-sets via functional reuse
Romdhane, M.S.B.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on ,Volume: 2 , 7-10 May 1996
Pages:1236 - 1239 vol. 2
19. A constructive deconvolution procedure of bandpass signals by homomorphic analysis
Marenco, A.L.; Madiseti, V.K.;
Geoscience and Remote Sensing Symposium, 1996. IGARSS '96. 'Remote Sensing for a Sustainable Future.', International ,Volume: 3 , 27-31 May 1996
Pages:1592 - 1596 vol.3

20. BEEHIVE: an adaptive, distributed, embedded signal processing environment
Famorzadeh, S.; Madiseti, V.; Egolf, T.; Nguyen, T.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE
International Conference on ,Volume: 1 , 21-24 April 1997
Pages:663 - 666 vol.1
21. Target detection from coregistered visual-thermal-range images
Perez-Jacome, J.E.; Madiseti, V.K.;
Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE
International Conference on ,Volume: 4 , 21-24 April 1997
Pages:2741 - 2744 vol.4
22. Variable block size adaptive lapped transform-based image coding
Klausutis, T.J.; Madiseti, V.K.;
Image Processing, 1997. Proceedings., International Conference on ,Volume:
3 , 26-29 Oct. 1997
Pages:686 - 689 vol.3
23. A Rate 8/10 (0, 6) MTR Code And Its Encoder / decoder
Jaejin Lee; Madiseti, V.K.;
Magnetics Conference, 1997. Digests of INTERMAG '97., 1997 IEEE
International , 1-4 April 1997
Pages:BS-15 - BS-15
24. VHDL models supporting a system-level design process: a RASSP
approach
DeBardelaben, J.A.; Madiseti, V.K.; Gadiant, A.J.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:183 – 188
25. A performance modeling framework applied to real time infrared
search and track processing
Pauer, E.K.; Pettigrew, M.N.; Myers, C.S.; Madiseti, V.K.;
VHDL International Users' Forum, 1997. Proceedings , 19-22 Oct. 1997
Pages:33 – 42
26. System design and re-engineering through virtual prototyping: a
temporal model-based approach
Khan, M.H.; Madiseti, V.K.;
Signals, Systems & Computers, 1998. Conference Record of the Thirty-
Second Asilomar Conference on ,Volume: 2 , 1-4 Nov. 1998
Pages:1720 - 1724 vol.2

27. A debugger RTOS for embedded systems
Akgul, T.; Kuacharoen, P.; Mooney, V.J.; Madiseti, V.K.;
Euromicro Conference, 2001. Proceedings. 27th , 4-6 Sept. 2001
Pages:264 - 269
28. Adaptability, extensibility and flexibility in real-time operating systems
Kuacharoen, P.; Akgul, T.; Mooney, V.J.; Madiseti, V.K.;
Digital Systems, Design, 2001. Proceedings. Euromicro Symposium on , 4-6 Sept. 2001
Pages:400 – 405
29. Effect of handoff delay on the system performance of TDMA cellular systems
Turkboylari, M.; Madiseti, V.K.;
Mobile and Wireless Communications Network, 2002. 4th International Workshop on , 9-11 Sept. 2002
Pages:411 – 415
30. Enforcing interdependencies and executing transactions atomically over autonomous mobile data stores using SyD link technology
Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on , 19-22 May 2003
Pages:803 – 809
31. Performance evaluation and optimization of SCTP in wireless ad-hoc networks
Argyriou, A.; Madiseti, V.;
Local Computer Networks, 2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on , 20-24 Oct. 2003
Pages:317 - 318
32. Implementation of a calendar application based on SyD coordination links
Prasad, S.K.; Bourgeois, A.G.; Dogdu, E.; Sunderraman, R.; Yi Pan; Navathe, S.; Madiseti, V.;
Parallel and Distributed Processing Symposium, 2003. Proceedings. International , 22-26 April 2003
Pages:8 pp.
33. Bandwidth aggregation with SCTP
Argyriou, A.; Madiseti, V.;
Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE Volume: 7 , 1-5 Dec. 2003
Pages:3716 - 3721 vol.7

34. Software streaming via block streaming
Kuacharoen, P.; Mooney, V.J.; Madisetti, V.K.;
Design, Automation and Test in Europe Conference and Exhibition, 2003
, 2003
Pages:912 – 917
35. Frequency-dependent space-interleaving for MIMO OFDM systems
Mohajerani, P.; Madisetti, V.K.;
Radio and Wireless Conference, 2003. RAWCON '03. Proceedings , Aug. 10-13, 2003
Pages:79 - 82
36. A media streaming protocol for heterogeneous wireless networks
Argyriou, A.; Madisetti, V.;
Computer Communications, 2003. CCW 2003. Proceedings. 2003 IEEE 18th Annual Workshop on , 20-21 Oct. 2003
Pages:30 – 33
37. Realizing load-balancing in ad-hoc networks with a transport layer protocol
Argyriou, A.; Madisetti, V.;
Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE ,Volume: 3 , 21-25 March 2004
Pages:1897 - 1902 Vol.3
38. Streaming H.264/AVC video over the Internet
Argyriou, A.; Madisetti, V.;
Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE , 5-8 Jan. 2004
Pages:169 – 174

Other Publications

1. A Transport Layer Technology for Improving the QoS of Networked Multimedia Applications <draft-madisetti-arguriou-qos-sctp-00.txt>.
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Jul 25, 2002.
2. Voice & Video over Mobile IP Networks <draft-madisetti-argyriou-voice-video-mip-00.txt>
Madisetti, V., Argyriou, A.
IETF Internet-Draft, Nov 20, 2002.
3. Enhancements to ECRTTP with Applications to Robust Header Compression for Wireless Applications. <draft-madisetti-rao-suresh-rohc-00.txt>
Madisetti, V.; Rao, S., Suresh, N.
IETF Internet-Draft, June 30, 2003.

Ph.D. Students Graduated

1. Brian T. Kelley, 1992
VLSI Computing Architectures for High Speed Signal Processing
Member of Technical Staff, Motorola.

Winner of Dr. Thurgood Marshall Dissertation Fellowship Award
2. Bryce A. Curtis, 1992
Special Instruction Set Multiple Chip Computer for DSP
Member of Technical Staff, IBM
3. Jaejin Lee, 1994
Robust Multitrack Codes for the Magnetic Channel
Professor, Yonsei University, Korea
4. Mohamed S. Ben Romdhane, 1995
Design Synthesis of Application-Specific IC for DSP
Director of IP, Rockwell.
5. Shoab A. Khan, 1995
Logic and Algorithm Partitioning on MCMs
Professor, National University of Science & Technology, Pakistan
6. Lan-Rong Dung, 1997
VHDL-based Conceptual Prototyping of Embedded DSP Architectures
Professor, National Chaio Tung University, Taiwan.

Winner of VHDL International Best PhD Thesis Award, 1997

7. Thomas W. Egolf, 1997
Virtual Prototyping of Embedded DSP Systems
Distinguished Member of Technical Staff, Agere
8. Alvaro Marenco, 1997
On Homomorphic Deconvolution of Bandpass Signals
Professor, Texas A&M University.

Winner of GIT ECE Outstanding Teaching Assistant Award

9. Shahram Famorzadeh, 1997
BEEHIVE: A Distributed Environment for Adaptive Signal Processing
Member of Technical Staff, Rockwell.
10. Timothy J. Klausutis, 1997
Adaptive Lapped Transforms with Applications to Image Coding.
US Air Force/Univ. of Florida.
11. Lan Shen, 1998
Temporal Design of Core-Based Systems
Member of Technical Staff, IBM
12. James DeBardelaben, 1998
Optimization Based Approach to Cost Effective DSP Design
Research Scientist, Johns Hopkins University

Georgia Tech ECE Faculty Award

13. Sangyoun Lee, 1999
Design of Robust Video Signal Processors
Professor, Yonsei University

US Army Sensors Lab Research Excellence Award, 1999

14. Rahmi Hezar, 2000
Oversampled Digital Filters
Member of Technical Staff, Texas Instruments
15. Yong-kyu Jung, 2001
Model-Based Processor Synthesis
Professor, Texas A&M University
16. Mustafa Turkboylari, 2002
Handoff Algorithms for Wireless Applications
Member of Technical Staff, Texas Instruments
17. Yun-Hui Fan, 2002
A Stereo Audio Coder with Nearly Constant Signal to Noise Ratio
Post-Doctoral Research Associate, Northeastern University
18. Subrato K. De, 2002

Design of a Retargetable Compiler for DSP
Member of Technical Staff, Qualcomm

US Army Sensors Lab Research Excellence Award, 1999

19. Chonlameth Aripnikanondt, 2004
System-on-Chip Design with UML
Professor, King Mongkut's University, Thailand.

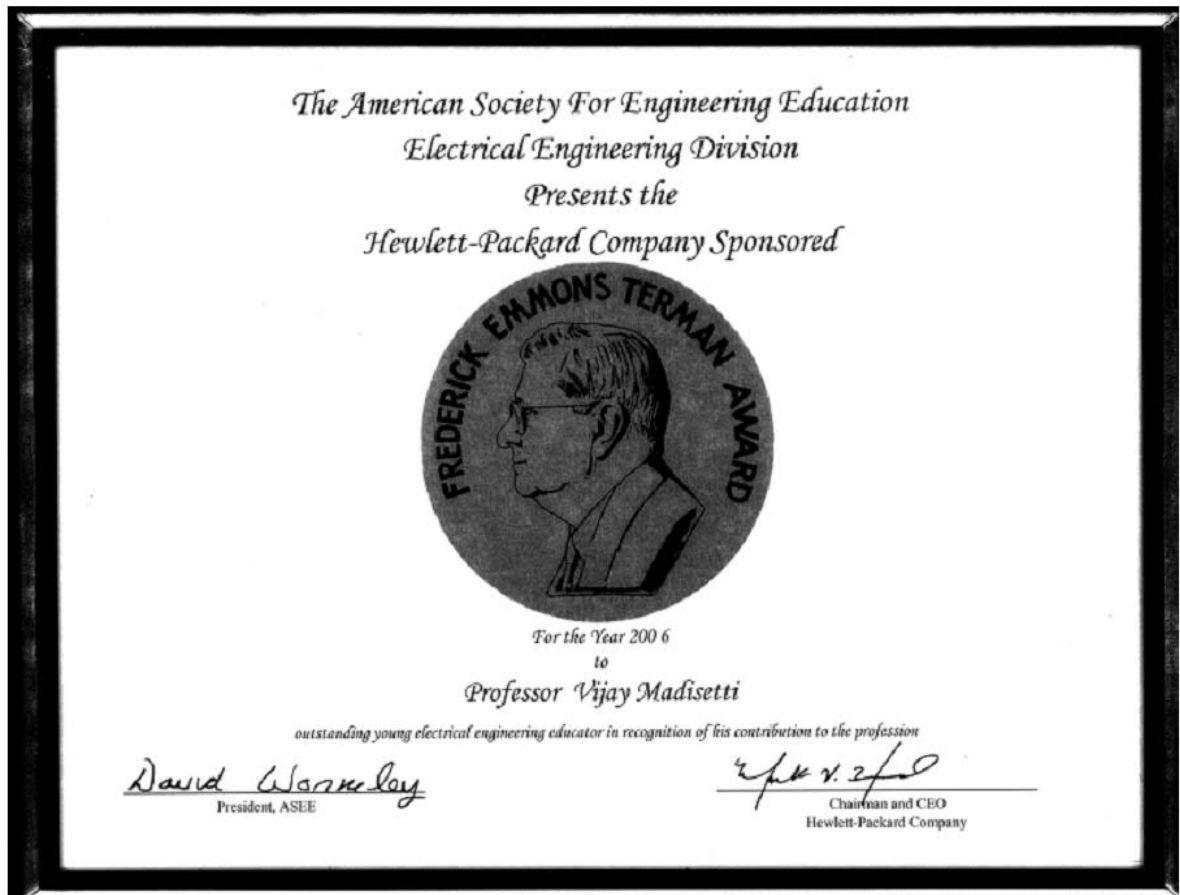
US Army Sensors Lab Research Excellence Award, 1999

20. Loran Jatunov, 2004
Performance Analysis of 3G CDMA Systems
Senior Research Scientist, Soft Networks, LLC.
21. Antonios Argyriou, 2005, Serving in Hellenic Army.
22. Pilho Kim, 2009, Scientist, VP Technologies, Inc.
23. M. Sinnokrot, 2009, Staff Engineer, Qualcomm.

Awards & Honors

1. **Jagadis Bose National Science Talent Fellowship**, Indian Institute of Technology, Kharagpur, 1980-1984.
2. **General Proficiency Prize**, Indian Institute of Technology, Kharagpur, 1984.
3. **Demetri Angelakos Outstanding Graduate Student Award**, Univ. of California, Berkeley, 1989
4. **Ira Kay IEEE/ACM Best Paper Award** for Best Paper presented at IEEE Annual Simulation Symposium, 1989
5. **IBM Faculty Development Award** 1990
6. **Technical Program Chair**, IEEE Workshop on Parallel and Distributed Simulation. 1990.
7. **Technical Program Chair**, IEEE MASCOTS'94
8. **NSF RI Award**, 1990
9. **VHDL International Best PhD Dissertation Advisor Award**, 1997

10. Georgia Tech Outstanding Doctoral Dissertation Advisor Award, 2001.
11. ASEE 2006 Frederick Emmons Terman Medal, 2006.
12. Fellow of IEEE



Intellectual Property Disclosures (Georgia Tech)

<u>Patent</u>	<u>Date</u>	<u>Description</u>
2843	2004	Method and Apparatus for Improving the Performance of Wireless LANs
2825	2003	Method and Apparatus for Optimal Partitioning and Ordering of Antennas for Layered Space-Time Block Codes in MIMO Communications Systems
2815	2003	How to Rapidly Develop a SyD Application
GSU-023	2003	Rapid Development of SyD Applications

2810	2003	System on Mobile Devices Middleware Design
2718	2003	A Transport Layer Algorithm for Improved Anycast Communication
2717	2003	A Novel Transport Layer Load-Balancing Algorithm
2716	2003	A Transport Layer QoS Algorithm
2715	2003	A Novel Transport Layer Algorithm for MPLS Performance
2659	2002	A New Algorithm and Technology for Implementing Mobile IP with Applications to Voice and Video over Mobile IP
2656	2002	Debugging with Instruction-Level Reverse Execution
2655	2002	Embedded Software Streaming
2539		System of Databases: An Enabling Technology for Programming
2517	2002	A Dynamic Instantiated Real-Time Operating System Debugger
2516	2002	A Dynamic Real-Time Operating System
GSU-009	2001	System of Databases: Architecture,, Global Queries, Triggers and Constraints
2480	2001	Mobile Fleet Application based on SyD Technology
2479	2001	System of Databases: A model with coordination links and a calendar application
1893	1999	Beehive
1726	1995	Very High Scale Integrated Circuit Hardware Description Language Models (VHDL Models)
1401	1995	Self-Compensation Receiver (SCR)

Issued Patents: 9,935,772, 9,769,213