

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

IRIUSRISK, INC.,
Petitioner

v.

THREATMODELER SOFTWARE INC.,
Patent Owner

IPR2023-00656

U.S. Patent No. 10,713,366

**DECLARATION OF ROBERT HURLBUT IN SUPPORT OF PETITION
FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 10,713,366**

TABLE OF CONTENTS

I.	INTRODUCTION	2
II.	BACKGROUND AND QUALIFICATIONS.....	2
III.	DOCUMENTS AND OTHER MATERIALS RELIED UPON.....	5
IV.	STATUTORY GROUNDS FOR CHALLENGE	6
V.	RELEVANT PATENT LAW AND LEGAL STANDARDS.....	6
	A. Date of Invention.....	6
	B. Anticipation	7
	C. Obviousness.....	7
	D. Standard of Proof.....	10
VI.	LEVEL OF ORDINARY SKILL IN THE ART	10
VII.	RELEVANT TECHNOLOGY BACKGROUND	11
VIII.	THE '366 PATENT AND ITS PROSECUTION HISTORY	15
	A. Overview of the '366 Patent.....	15
	B. Priority Date	19
	C. Prosecution History	19
IX.	CLAIM CONSTRUCTION.....	22
	A. “third party software application”	22
	B. “third party diagram components” and “visual representations of the threat model components”	23
	C. “threat model components”	23
X.	INVALIDITY OF THE CHALLENGED CLAIMS OF THE '366 PATENT	25
	A. Overview of the Prior Art.....	26
	1. Keenan (EX-1004).....	26
	2. Zheng (EX-1005).....	30
	3. Baker (EX-1006).....	32
	4. Jones (EX-1007)	34

5.	Galliano (EX-1008).....	34
B.	Ground 1: Keenan Anticipates the Challenged Claims.....	35
1.	Claim 1	35
2.	Claim 2: “The method of claim 1, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	62
3.	Claim 3: The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.....	63
4.	Claim 4: “The method of claim 1, wherein the data file comprises one of a VSD file, a VSDX file, a TMT file, an XML file, a JSON file, a PNG file, and a JPEG file.”	64
5.	Claim 5: “The method of claim 1, wherein determining, for the subset of third party diagram components, correlated threat model components, comprises first determining whether a user-defined name for one of the subset of third party diagram components has a match in the one or more mapping files and, if no match is found, then determining whether a default title for the third party diagram component, as defined by the third party software application, has a match in the one or more mapping files.”	64
6.	Claim 6: “The method of claim 5, further comprising, if no match is found for the default title, matching the third party diagram component with a placeholder component and displaying the placeholder component on the relational diagram to indicate that no match was found.”	66
7.	Claim 7: “The method of claim 1, further comprising determining, for the one or more relationships defined by the data file, correlated threat model components as defined in the one or more mapping files.”	68
8.	Claim 8:.....	69

9.	Claim 9: “The system of claim 8, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	76
10.	Claim 10: The system of claim 8, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.....	76
11.	Claim 11: “The system of claim 8, wherein the data file comprises one of a VSD file, a VSDX file, a TMT File, an XML file, a JSON file, a PNG file, and a JPEG file.”	76
12.	Claim 12: “The system of claim 8, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.”	76
13.	Claim 13: “The system of claim 8, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.”	77
14.	Claim 14: The system of claim 8, wherein, for each of the subset of third party diagram components for which no correlated threat model component is found in the one or more mapping files, the diagram interface displays a placeholder component indicative of a match not being found.	78
15.	Claim 15: The system of claim 8, further comprising one or more mapping interfaces displayed on the one or more computing devices and configured to, in response to receiving user input, modify the one or more mapping files.....	78
16.	Claim 16:.....	79
17.	Claim 17: “The system of claim 16, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	81

18.	Claim 18: The system of claim 16, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.....	81
19.	Claim 19: The system of claim 16, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.	81
20.	Claim 20: “The system of claim 16, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.”	82
C.	Ground 2: Zheng In View of Baker and Jones Renders Obvious Claims 1-4, 7-11, and 15-18.	82
1.	Claim 1	82
2.	Claim 2: “The method of claim 1, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	101
3.	Claim 3: The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.	102
4.	Claim 4: “The method of claim 1, wherein the data file comprises one of a VSD file, a VSDX file, a TMT file, an XML file, a JSON file, a PNG file, and a JPEG file.”	102
5.	Claim 7: “The method of claim 1, further comprising determining, for the one or more relationships defined by the data file, correlated threat model components as defined in the one or more mapping files.”	103
6.	Claim 8:	104
7.	Claim 9: “The system of claim 8, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	110

8.	Claim 11: The system of claim 8, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.....	110
9.	Claim 15: The system of claim 8, further comprising one or more mapping interfaces displayed on the one or more computing devices and configured to, in response to receiving user input, modify the one or more mapping files.....	110
10.	Claim 16:.....	112
11.	Claim 17: “The system of claim 16, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”	114
12.	Claim 18: The system of claim 16, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.....	114
13.	A POSITA Would Have Been Motivated to Combine Zheng with Baker and Jones.	114
D.	Ground 3: The References of Ground 2 in Further View of Galliano Render Obvious Claims 2, 9, and 17.....	117
E.	Ground 4: The References of Ground 2 in Further View of Keenan Render Obvious Claims 1-20	119
F.	Ground 5: The References of Ground 2 in Further View of the Knowledge of a POSITA Render Obvious Claims 2, 5, 6, 9, 12-14, 17, 19 and 20	120
XI.	CONCLUSION	122

EXHIBITS

Exhibit No.	Description
EX-1001	U.S. Patent No. 10,713,366 (“the ’366 patent”)
EX-1002	Prosecution History of the ’366 patent
EX-1003	Declaration of Robert Hurlbut
EX-1004	U.S. Patent No. 11,200,228 (“Keenan”)
EX-1005	U.S. Patent No. 10,503,907 (“Zheng”)
EX-1006	U.S. Patent Pub. 2014/0236665 (“Baker”)
EX-1007	U.S. Patent No. 9,602,529 (“Jones”)
EX-1008	U.S. Patent No. 10,681,068 (“Galliano”)
EX-1009	Federal Judiciary Report
EX-1010	Email from Petitioner’s Counsel to PO’s Counsel
EX-1011	Excerpts from <i>Risk Centric Threat Modeling</i> , by Tony Uceda Velez and Marco M. Morana, John Wiley and Sons, Inc., 2015
EX-1012	Excerpts from <i>Threat Modeling Designing for Security</i> , by Adam Shostack, John Wiley and Sons, Inc., 2014
EX-1013	Excerpts from <i>Threat Modeling</i> , by Frank Swiderski and Window Snyder, Microsoft Press, 2004
EX-1014	Excerpts from <i>The Security Development Lifecycle</i> , by Michael Howard and Steve Lipner, Microsoft Press, 2006

I. INTRODUCTION

I, Robert Hurlbut, declare as follows:

1. My name is Robert Hurlbut. I have been asked by Petitioner IriusRisk Inc. to provide my expert opinions in support of the above-identified petition for *inter partes* review of U.S. Patent No. 10,713,366 (the “’366 patent,” EX-1001), challenging the validity of claims 1-20 (the “Challenged Claims”).

2. I currently hold the opinions set forth in this declaration.

3. In summary, it is my opinion that the references cited below anticipate or render obvious the Challenged Claims. My detailed opinions on the claims are set forth below and in the attached claim chart.

4. I am being compensated for my time at the rate of \$250 per hour. This compensation is not contingent upon my performance, the outcome of this matter, or any issues involved in or related to this matter.

II. BACKGROUND AND QUALIFICATIONS

5. I have extensive experience in threat modeling and application security.

6. My professional experience and expertise is described in **Exhibit A**, which is my current CV.

7. I obtained a Bachelor of Science in Mathematics from Oklahoma State University in 1990.

8. I obtained a Master of Science in Cyber Security from Southern New Hampshire University in 2020.

9. During the 33 years after my undergraduate education, including while I was obtaining my Master's degree, I worked in industry as a software developer, software architect, and technical consultant for numerous companies that are listed in Exhibit A.

10. Since 2015, a significant amount of my work has been in designing and building threat models.

11. Between 2015 and the present, I have provided threat modeling consulting services for numerous clients. I design and build threat models for these clients and train their employees regarding the designing, building, and use of threat models. The threat models assist the clients in identifying potential risks to various portions of their businesses and infrastructure. This can include identifying potential threats in any applications or products the client may create as well as recommended mitigations to remediate those threats.

12. Between September 2017 and November 2021, I was the Senior Vice President and Threat Modeling Senior Architect/Lead at Bank of America, where I started, led, and further developed Bank of America's threat modeling program. I co-wrote materials for threat model training sessions that was required of all software developers within Bank of America. I worked with teams to build over

1,000 threat models within three years. These threat models identified potential threats to dozens of aspects of Bank of America's business areas, including threats to its storage and processing systems, its communication networks, its employees, its physical buildings, and personal information of customers.

13. Between November 2021 and the present, I have been the Principal Application Security Architect and Threat Modeling Lead at Aquia, Inc. I lead the threat modeling program at Aquia and for many of Aquia's client's, including a federal government agency.

14. I am a co-author of the Threat Modeling Manifesto, published in 2020, which was intended to inform, educate, and inspire practitioners to adopt threat modeling and improve security and privacy during the software development process. This is a "best practices" guide that my co-authors and I wanted to share with the public.

15. Since 2022, I am a co-Founding Member and one of the Speaker / Facilitators of a community endeavor called Threat Modeling Connect, which is an open forum to inform, educate, and inspire practitioners to adopt threat modeling and improve security and privacy during the software development process. This volunteer organization also provides community meetups, training events, and will host a first threat modeling conference in late 2023.

16. I have been a Certified Secure Software Lifecycle Professional (CSSLP) for the years 2014-2023 (renewed annually). This is a certification provided by the cybersecurity professional organization known as (ISC).

17. I am a frequent speaker at user groups and national and international conferences on threat modeling and application security.

III. DOCUMENTS AND OTHER MATERIALS RELIED UPON

18. In forming the opinions set forth in this declaration, I have reviewed the documents in the following table:

Exhibit No.	Description
EX-1001	U.S. Patent No. 10,713,366 (“the ’366 patent”)
EX-1002	Prosecution History of the ’366 patent
EX-1003	EX-1003 is this Declaration.
EX-1004	U.S. Patent No. 11,200,228 (“Keenan”)
EX-1005	U.S. Patent No. 10,503,907 (“Zheng”)
EX-1006	U.S. Patent Pub. 2014/0236665 (“Baker”)
EX-1007	U.S. Patent No. 9,602,529 (“Jones”)
EX-1008	U.S. Patent No. 10,681,068 (“Galliano”)
EX-1011	Excerpts from <i>Risk Centric Threat Modeling</i> , by Tony Uceda Velez and Marco M. Morana, John Wiley and Sons, Inc., 2015
EX-1012	Excerpts from <i>Threat Modeling Designing for Security</i> , by Adam Shostack, John Wiley and Sons, Inc., 2014
EX-1013	Excerpts from <i>Threat Modeling</i> , by Frank Swiderski and Window Snyder, Microsoft Press, 2004

Exhibit No.	Description
EX-1014	Excerpts from <i>The Security Development Lifecycle</i> , by Michael Howard and Steve Lipner, Microsoft Press, 2006

19. Additionally, I have utilized my own experience and expertise, including regarding the knowledge and capabilities of a person of ordinary skill in the relevant art in the timeframe of the claimed priority date of the '366 patent.

IV. STATUTORY GROUNDS FOR CHALLENGE

20. In this declaration, I offer opinions on the following grounds that are the basis for this IPR proceeding:

Ground(s)	Claims	Basis	Reference(s)
1	1-20	§102	Keenan
2	1-4, 7-11, 15-18	§103	Zheng in view of Baker and Jones
3	2, 9, and 17	§103	The references of Ground 2 in further view of Galliano
4	1-20	§103	The references of Ground 2 in further view of Keenan
5	2, 5-6, 9, 12-14, 17, 19-20	§103	The references of Ground 2 in further view of the knowledge of a POSITA

V. RELEVANT PATENT LAW AND LEGAL STANDARDS

A. Date of Invention

21. I have been asked to use the date of the filing of the earliest application to which priority is claimed (*i.e.*, May 17, 2017) as the date of invention for purposes of my analysis.

B. Anticipation

22. I am informed and understand that a patent claim is invalid under 35 U.S.C. §102 if a prior art disclosure teaches every element required by the claim, either expressly or inherently. The identical invention must be shown in as complete detail as is contained in the claim. When a claim covers several structures or compositions, either generically or as alternatives, the claim is deemed anticipated if any of the structures or compositions within the scope of the claim is known in the prior art.

C. Obviousness

23. I am informed and understand that a patent claim is invalid under 35 U.S.C. § 103 if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time of the invention to a person having ordinary skill in the art to which the subject matter pertains. Obviousness, as has been explained to me, is based on (i) the scope and content of the prior art, (ii) the differences between the prior art and the claim, (iii) the level of ordinary skill in the art, and (iv) any secondary indicia of non-obviousness (*e.g.*, “secondary considerations” such as commercial success in the market place of the claimed invention), to the extent that they exist.

24. I further understand that obviousness may be demonstrated by showing that it would have been obvious to combine the teachings of more than

one item of prior art. 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, the following are examples of approaches and rationales that may be considered:

- Combining prior art elements according to known methods to yield predictable results;
- Simple substitution of one known element for another to obtain predictable results;
- Use of a known technique to improve similar devices (methods, or products) in the same way;
- Applying a known technique to a known device (method or product) ready for improvement to yield predictable results;
- 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);
- Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to a person having ordinary skill in the art; and

- Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

25. I further understand that an invention may be obvious if one of ordinary skill in the art, facing a wide range of needs created by developments in the field, would have seen an obvious benefit to the solutions tried by the applicant. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, it may be obvious to a person of ordinary skill to try the known options. If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique would have been obvious.

26. I further understand that a person of ordinary skill in the art is a hypothetical person who is presumed to be aware of all of the relevant art at the time of the invention. The person of ordinary skill may be able to fit together the teachings of multiple patents and/or printed publications employing ordinary creativity and the common sense that familiar items may have uses in another context or beyond their primary purposes.

27. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists so as to avoid impermissibly applying hindsight when combining or modifying the prior art. I understand this test should not be applied rigidly, but that the test can be important to avoid such hindsight.

28. I also understand that one of ordinary skill in the art must have a reasonable expectation of success in combining or modifying prior art references.

29. I also understand that all elements of a claim must be considered in an obviousness analysis.

D. Standard of Proof

30. I understand that the standard to prove unpatentability in an *inter partes* review proceeding is by a preponderance of the evidence, which means more likely than not.

VI. LEVEL OF ORDINARY SKILL IN THE ART

31. I have considered the level of ordinary skill in the art relevant to the '366 patent. I understand that factors such as the education level of those working in the field, the sophistication of the technology, the types of problems encountered in the art, the prior art solutions to those problems, and the speed at which innovations are made may help establish the level of skill in the art.

32. After reviewing the '366 patent, the relevant portions of the prosecution history of the '366 patent, and cited references, it is my opinion that the relevant art of the '366 patent is in the field of threat modeling.

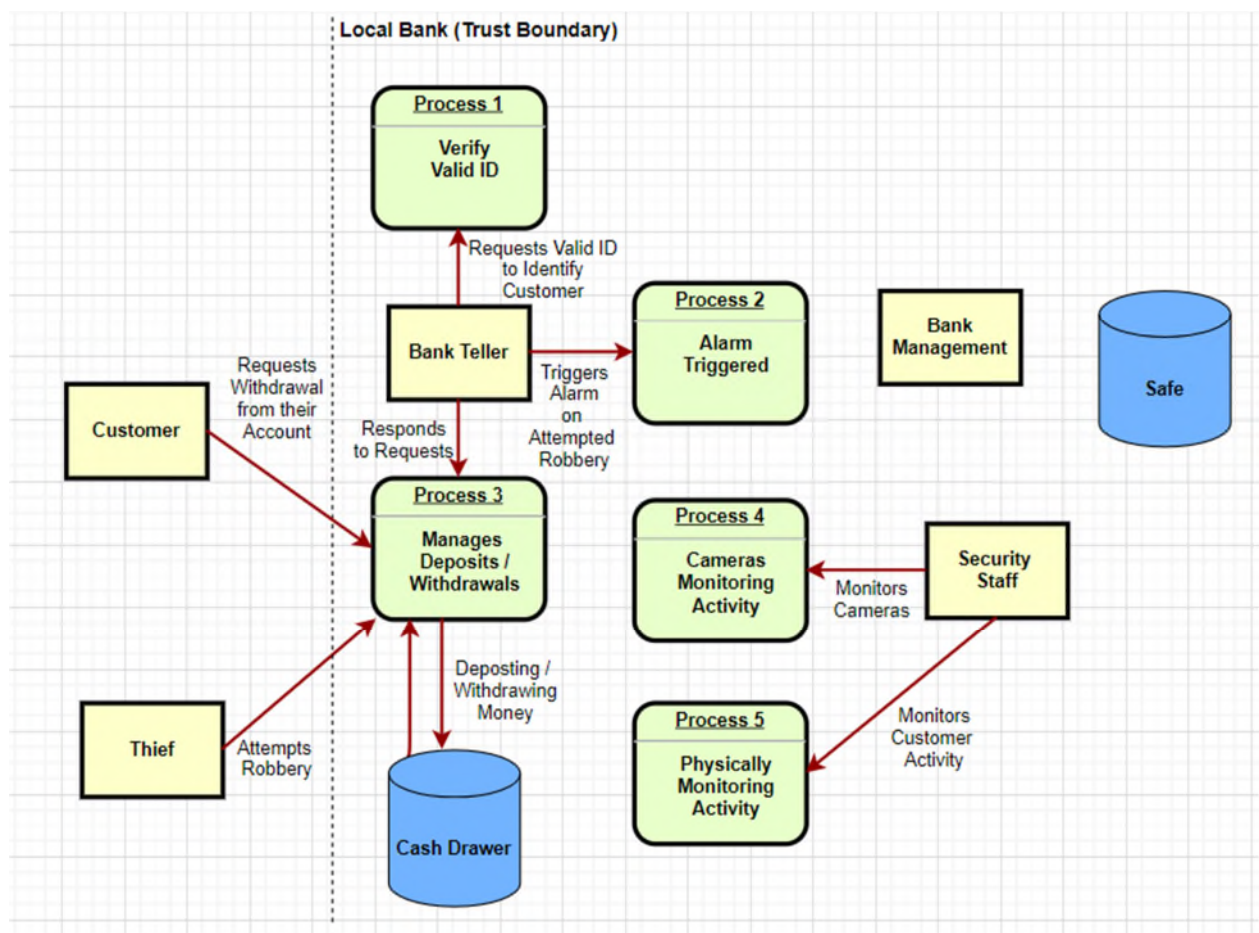
33. Based upon my knowledge of this field, I conclude that a person of ordinary skill in the art ("POSITA") at the time of the earliest filing date of the '366 patent (May 17, 2017), and for that matter, at all subsequent times through the present, would have had a Bachelor's degree in computer engineering, computer science, mathematics, or a similar discipline, with at least three years of relevant industry or research experience, including experience designing or implementing threat models and experience designing or developing software applications. Additional work or research experience can substitute for less or different education, and vice-versa.

34. I am and was at the time of the earliest filing date at least a person of skill in the relevant art, of a level at least as high as or more than ordinarily skilled persons, and I am able to provide insight and opinions into the understandings of such persons at all relevant times by virtue of my education, training and experience.

VII. RELEVANT TECHNOLOGY BACKGROUND

35. Threat modeling is a process by which potential threats to a system are identified. The system can be almost anything that is vulnerable to threats and

that humans wish to protect. For example, it can be a physical structure, such as an office building or a power plant, or it can be a digital asset, such as financial data or customer information. A threat model is a useful tool for human beings to systematically understand a system's vulnerabilities. Historically, threat models were often developed by people drawing on a white board or a piece of paper. A simple example of a threat model is the following, which shows potential threats to physical branch of a bank:



In this diagram, called a Data Flow Diagram (DFD), which is commonly used for Threat Modeling, the following elements are depicted:

- Actors: Customer, Thief, Bank Teller, Bank Management, Security Staff
- Processes: Verify Valid ID, Manages Deposits / Withdrawals, Alarm Triggered, Camara Monitoring Activity, Physically Monitoring Activity
- Data Stores or Assets: Cash Drawer, Safe
- Data Flows: Arrows in the diagram to show actions or data flowing between Actors and Processes or between Processes and Data Stores or Assets

In determining threats, this Data Flow Diagram (DFD) helps identify potential threat actors (Customer, Thief) and various countermeasures to create an initial threat model:

1. Threat: Customer requests the Bank Teller to withdraw cash from their Account
 - a. Countermeasure(s):
 - i. Bank Teller requests Valid ID
 - ii. Bank Teller verifies Valid ID

2. Threat: Thief attempts robbery by requesting the Bank
Teller to withdraw cash from the Cash Drawer

a. Countermeasure(s):

- i. Bank Teller triggers an Alarm
- ii. Security Staff monitors the camera for
suspicious activity
- iii. Security Staff monitors actions by customers
for suspicious activity

36. It is common for software developers to utilize threat models during the software development process so that they can identify potential risks and build in safeguards against those risks in the software as they are developing it. This is discussed, for example, in the well-known treatise *Risk Centric Threat Modeling*, by Tony Uceda Velez and Marco M. Morana, John Wiley and Sons, Inc., 2015, which states, “There are several good reasons to integrate threat modeling activities within the SDLC (software development lifecycle). Threat models help information security reviews of software projects and assess the risks posed by vulnerabilities identified during the various type security activities that are built into SDLC such as design reviews, secure coding, and penetration tests. A threat model help risk managers to assess application security risks by correlating the analysis of threats to the vulnerabilities to application data and functional assets. A

risk analysis that factors threats, vulnerabilities, and assets can also factor likelihoods and impacts to determine the risk level and to apply security measures to reduce this level of risk.” EX-1011 (excerpts of treatise), p. 94. Many other treatises from the early 2000s reiterate the commonality of different threat models. EX-2012, *Threat Modeling Designing for Security*, by Adam Shostack, John Wiley and Sons, Inc., 2014 (“Everyone threat models.” Page xxii); EX-2013, *Threat Modeling*, by Frank Swiderski and Window Snyder, Microsoft Press, 2004 (“The process of threat modeling is not new.” Page xi); EX-1014, *The Security Development Lifecycle*, by Michael Howard and Steve Lipner, Microsoft Press, 2006 (“Chapter 9: Threat Modeling Artifacts.”).

37. In the late 1990s, computer software emerged for preparing threat models. For example, Microsoft introduced the SDL Threat Modeling Tool that allowed users to build threat models using a computer running the tool. I personally used the Microsoft tool for the first time around 2009.

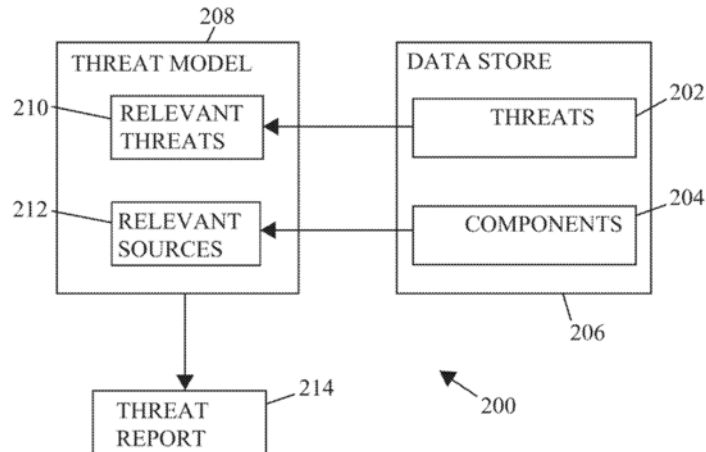
VIII. The '366 PATENT AND ITS PROSECUTION HISTORY

A. Overview of the '366 Patent

38. The '366 patent, entitled “Systems and methods for automated threat model generation from third party diagram files,” relates to threat modeling processes and systems. EX-1001, cover page. The patent explains that “[t]hreat modeling is a process by which vulnerabilities of a system or process may be

detailed and prioritized.” *Id.*, 1:56-57. Threat modeling was well-known in the prior art and involves the identification and visualization of potential threats, which typically involves generating diagrams on a whiteboard or in a computer application to represent potential threats to a particular system. The ’366 patent explains that “[t]hreat modeling allows a user to analyze potential attack vectors and prioritize vulnerabilities. While some threat modeling involves threats related to computing networks and systems, threat modeling in general encompasses a broader scope and may involve modeling threats in non-computer-related systems and processes.” *Id.*, 1:60-66. The patent explains that examples of threat models could include modeling the possible threats to preventing the spread of an infectious disease, or modeling the possible attacks on a computer network (cybersecurity).” *Id.*, 9:1-5.

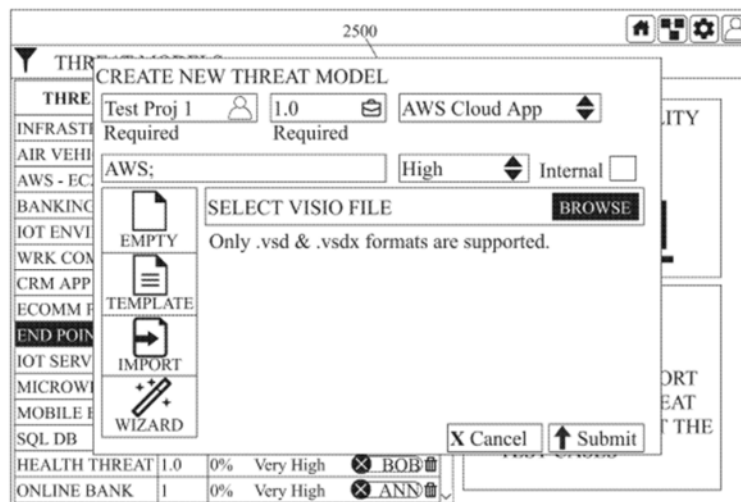
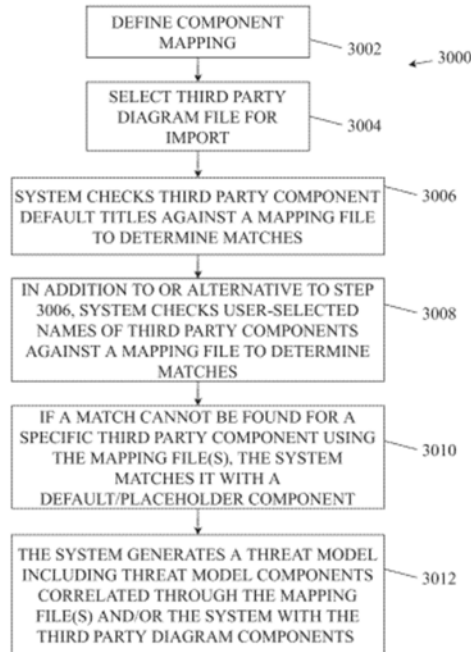
39. The ’366 patent describes that threat models are created from defined threats and threat model components, which are stored within a database. *Id.*, 9:10-16 (“As illustrated in FIG. 2, process 200 may include storing a plurality of threats 202 and threat model components (components) 204 in a data store 206. This may include, by non-limiting example, storing titles, definitions or descriptions, and/or associated images in the database 108 for each component and/or threat. The threats and components are used to create threat model 208.”). This is shown in Figure 2:



40. Each threat model component “is a potential relevant source for one or more threats.” *Id.*, 9:55-10:7.

41. The ’366 patent further describes creating visual diagrams in a third party software application and using those diagrams in the threat model. *Id.*, 24:26-29. Specifically, the patent describes a mapping file stored within the database that correlates threat model components with visual diagram components of a “third party” software application, such as Microsoft Visio. *Id.*, 37:14-18. *Id.*, 37:25-39.

42. A visual diagram created in Microsoft Visio or other third party application may be imported for use in the threat model. “The threat modeling system and methods include the modeling of threats utilizing software which users access and interact with through a variety of user interfaces.” *Id.*, 8:59-64. See also Figures 25 and 30



43. The threat model system further “is used to generate an original threat report (report) 214 which in implementations includes identified threats, the status of identified threats (threat status), and the source(s) of identified threats, among other things.” *Id.*, 9:6-9.

44. The patent owner (“PO”) was not the first to devise the claimed invention, as shown through the prior art discussed below—none of which was before the Patent Office during prosecution.

B. Priority Date

45. The ’366 patent claims priority to four provisional applications, the earliest of which was filed on May 17, 2017. For the purposes of this IPR only, Petitioner applies a priority date of May 17, 2017 to this IPR.

C. Prosecution History

46. The ’366 patent was filed as U.S. Patent App. No. 16/542,263 on August 15, 2019 with 20 pending claims. EX-1002, Application. On November 18, 2019, the Examiner issued a non-final rejection against all pending claims under 35 U.S.C. § 112(b) as being indefinite. *Id.*, Office Action at pp. 3-6. In response, Applicant and Examiner participated in an interview on January 7, 2020, during which Applicant clarified the meaning of the terms that led to the rejection, as described below. *Id.*, Applicant Initiated Interview Request Form, Response to Office Action dated November 18, 2019, p. 8.

47. First, in the Office Action the Examiner rejected independent claims 1, 8, and 16 noting that “a person of ordinary skill in the art would not be able to determine how the claimed ‘visual representations’ relate to the ‘third party diagram components,’” where the third party diagrams themselves are visual

representations. *Id.*, Office Action at p. 4. During the interview, Applicant clarified that the “third party diagram components” and “visual representations of the threat model components” were distinct elements. *Id.*, Response to Office Action dated November 18, 2019 at p. 8. Specifically, Applicant asserted that “the third party diagram components [are] components of an independent software application and the visual representations of the threat model components [are] components of the native software, stored in databases/servers locally or remotely but coupled with the system.” *Id.*, Response to Office Action dated November 18, 2019 at p. 8. The Applicant further stated:

The mapping file correlates the internal threat model components with the external third party diagram components so that the system can read a file created with the third party software program and generate its own diagram using internal components that are, through the mapping file, correlated to the external third party components.

Id., Response to Office Action dated November 18, 2019 at p. 9.

48. Second, in the Office Action the Examiner rejected claim 16 because it contained the trademark “VISIO.” *Id.*, Office Action at pp. 4-5. Applicant removed the “VISIO” term from the claims and this rejection was withdrawn. *Id.*, Response to Office Action dated November 18, 2019 at p. 9.

49. Third, in the Office Action the Examiner rejected claims 1, 5, 6, 8, 12-14, 19-20 for uncertainty due to use of the term “third party components,” noting that “third party” “refers to an issue of ownership, which can change at arbitrary

times.” *Id.*, Office Action at pp. 5-6. During an interview, Applicant and Examiner agreed that “third party” would not necessarily indicate present ownership, but rather “independently-created software applications not created with a common schema, such that a mapping file or the like as claimed in the claims would be useful to map between elements of the two software applications.” *Id.*, Response to Office Action dated November 18, 2019 at pp. 9-10. Applicant further provided a number of examples to illustrate the meaning of “third party.” *Id.*, Response to Office Action dated November 18, 2019 at pp. 10-11.

50. Based on statements made by the Applicant during the interview and associated claim amendments, the Examiner withdrew all rejections and issued a notice of allowance on March 5, 2020. *Id.*, Notice of Allowance. As reasons for allowance, the Examiner noted that the prior art of record, particularly U.S. Pub. No. 2015/0033346 (“Herbert”) and U.S. Patent No. 8,413,249 (“Chou”) did not disclose, with respect to claim 1, “determining using the one or more computing devices, for the subset of third party diagram components, correlated threat model components as defined in the one or more mapping files.” *Id.*, Notice of Allowance at pp. 2-3. Moreover, the Examiner found that neither Chou nor Herbert “discloses third-party control of the diagram components of the architecture model, wherein ‘the one or more mapping files correlating the threat model components with visual diagram components of a third party software

application.” *Id.* The Examiner found that the same reasoning applied to the similar elements found in the remaining independent claims (8 and 16) and thus all claims were allowable. *Id.*

51. Thus, claim limitations regarding importing a diagram from a third party software application was an important basis for the Examiner’s allowance.

IX. CLAIM CONSTRUCTION

52. In my opinion, the ’366 patent and its file history establish the following definitions for certain terms in the Challenged Claims, and I have applied those definitions in my analysis. For all other terms, I applied the plain and ordinary meaning that a POSITA would have for the terms.

A. “third party software application”

53. As discussed in a previous section, the Examiner rejected the application because, among other reasons, the term “third party” was indefinite. EX-1002, Office Action at pp. 5-6. To overcome this rejection, Applicant asserted during an Examiner interview that “third party” did not necessarily indicate ownership, and instead expressly defined “third party applications” to mean “independently-created software applications not created with a common schema, such that a mapping file or the like as claimed in the claims would be useful to map between elements of the two software applications.” *Id.*, Response to Office Action dated November 18, 2019 at pp. 9-10. Applicant further made clear

through the use of examples that commonly-owned software can still be “third party” software where the applications are not created with a common schema and thus the use of a mapping file as claimed would still be useful. *Id.*, Response to Office Action dated November 18, 2019 at pp. 10-11.

54. Based on these statements in the file history, “third party software application” means “independently-created software applications not created with a common schema, such that a mapping file or the like as claimed in the claims would be useful to map between elements of the two software applications.”

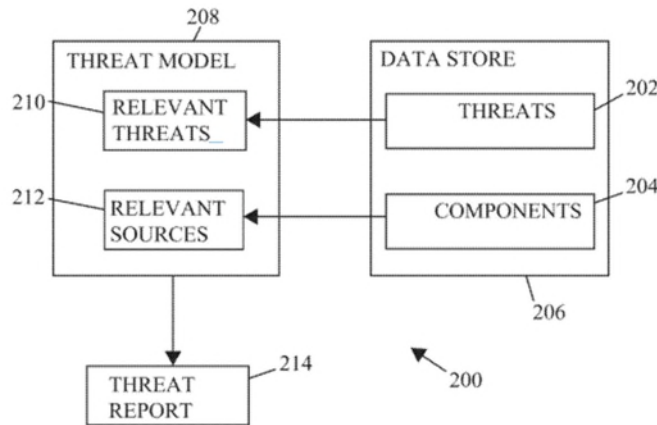
B. “third party diagram components” and “visual representations of the threat model components”

55. The applicant provided further definitions in the file history, asserting that “the third party diagram components [are] components of an independent software application,” and that “visual representations of the threat model components [are] components of the native software, stored in databases/servers locally or remotely but coupled with the system.” *Id.*, Response to Office Action dated November 18, 2019 at p. 8.

C. “threat model components”

56. The ’366 patent specification explains that “[a]s illustrated in FIG. 2, process 200 may include storing a plurality of threats 202 and threat model components (components) 204 in a data store 206. This may include, by non-limiting example, storing titles, definitions or descriptions, and/or associated

images in the database 108 for each component and/or threat. The threats and components are used to create threat model 208.” ’366 patent, 9:10-16.

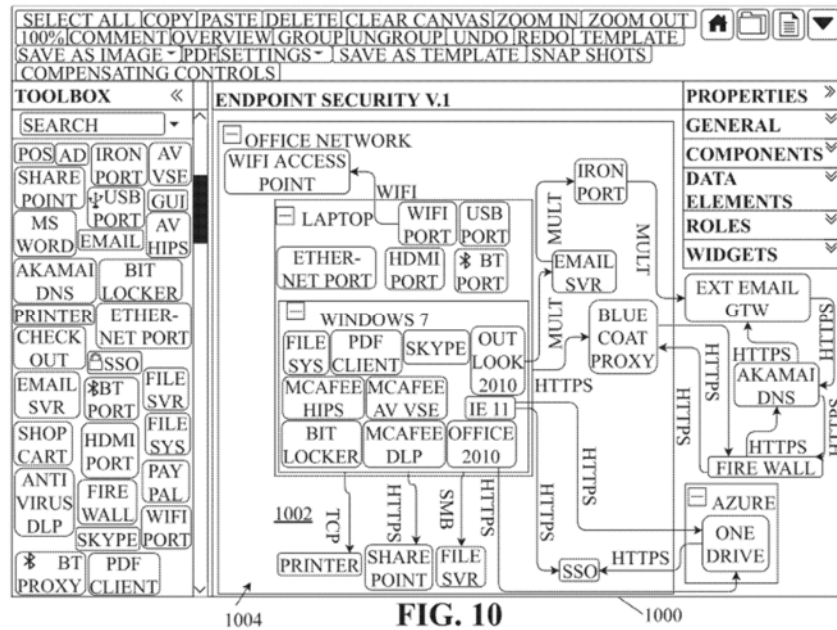


Id., Fig. 2.

57. In discussing Figure 2, the patent explains that a threat model “component” is “a potential relevant source for one or more threats.” *Id.*, 9:55-57.

58. The specification further explains that the threat model components are not limited to physical components of a computing device or system/network but may also be action steps or any other source of threats. *Id.*, 9:65-10:13.

59. Figure 10 shows an exemplary user interface of the system described by the ’366 patent where other example components are included in the toolbox on the left side of the interface:



Id., Fig. 10. The specification further explains, “The components displayed in the toolbox of FIG. 10 are generally shown without icons. The USB Port, SSO (single sign on), BLUETOOTH PORT, and BLUETOOTH PROXY components are shown with icons as an example that all components may be shown in the toolbox with an icon if desired.” EX-1001, 17:36-41.

60. In my opinion, the specification indicates that the term “threat model component” means “a potential relevant source for one or more threats.”

X. INVALIDITY OF THE CHALLENGED CLAIMS OF THE '366 PATENT

61. In my opinion, the Challenged Claims are invalid for the reasons discussed below and in the attached Exhibit B (which is an anticipation chart for Keenan).

A. Overview of the Prior Art

1. Keenan (EX-1004)

62. U.S. Patent No. 11,200,228 (“Keenan”), entitled, “Integrated object environment system and method,” was filed on April 19, 2017, and issued on December 14, 2021. I am informed that Keenan is prior art to the ’366 patent under 35 U.S.C. § 102(a)(2).

63. Keenan discloses “an Integrated Object Environment (IOE) running in a graph database environment.” EX-1004, Abstract. Keenan’s threat modeling system and method “include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database.” *Id.* The IOE “stores, reveals and links characteristics (Structural Metadata) of each object in the topology either within the IOE or in a linked repository, such that accountability, benefits and consequences of using those assets can easily and repeatedly be measured in the cost and profitability of the outcomes those assets enable.” *Id.*, 1:47-52. A user may interact with the IOE through a Graphical User Interface (GUI), including “a View mode, where the user can view any section of the topology and Characteristics by Object” and “a Modify mode, where the user can add, change and delete Objects or Characteristics.” *Id.*, 11:42-51. “The intelligence embedded in the stored topology can be viewed in a variety of ways through the GUI, both in terms of the object-to-object relationships

and the associated characteristics of objects.” *Id.*, 12: 64-67. One such manner of viewing the IOE through the GUI is by processing and viewing threat vectors, which “fills the object shapes based on the value of the technical risk ranking characteristics of that object.” *Id.*, 16:21-23, 14:7-15:10. Figure 16 shows one such example:

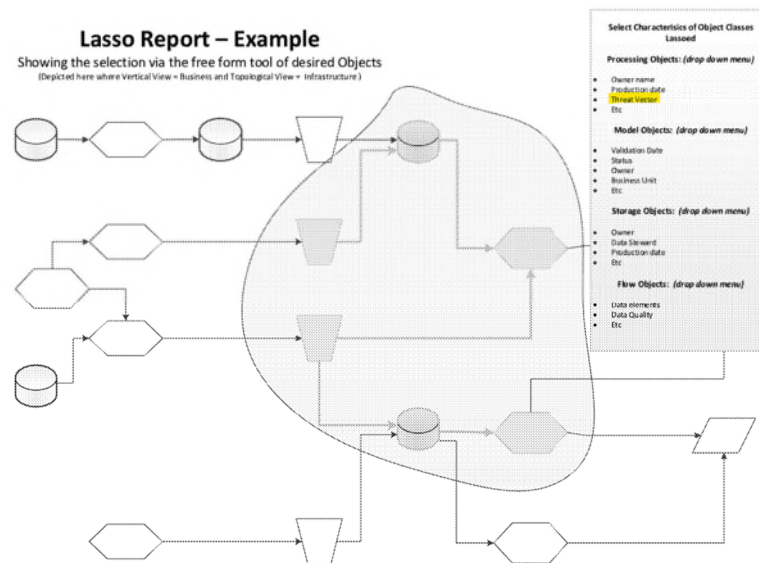


FIGURE 16

Keenan further explains its variety of views, including threat vectors, by stating that “[t]he GUI preferably permits user selection of other options to view the topology, which can be selected in combination with one another. A summary of View options, in accordance with an embodiment, is shown in Table 3. *Id.*, 14:7-15:10.

TABLE 3

View Name	Options	Description
OBJECT CLASS	All Classes Specify Object Classes (Flow objects will not be shown unless all object classes are selected)	All object classes are depicted User can select one or more classes to depict. Data Flow Lines will not be depicted unless all classes are selected. In lieu of data flow lines, objects will be joined by lines that look like data flow lines but the arrow heads will be a different shape.
HORIZONTAL	Linear Network	An instance of a particular Data Storage object appears for every extract and load downstream of the initial instance of the Data Storage object in the View all Data Storage objects appear only once in the View
VERTICAL	Business IT Architecture IT Infrastructure	Highest level functionality Application centric Hardware centric
PALETTE	Platform Model Risk Data Quality Risk Capital Allocation Business Entity Business Owner Value by Business Metric Use Age Technical Threat Vector Operational Risk	color fills the object shapes based on the platform that hosts the object. Objects not hosted by platforms will have white fill. color fills the object shapes based on the value of the risk characteristics for the models in the view color fills the Flow arrowhead shapes based on the value of the value of data quality characteristic for the data elements feeding the object. This palette can be selected color fills the object shapes to indicate the absolute value of Risk capital allocated to the object. color fills the object shapes based on the Business Entity or Entities that use the output from a Path of objects color fills the object shapes based on the business entity that owns or co-owns the object color fills the object shapes to differentiate the value of the business metric(s) associated with each production chain of objects in the view. color fills the object shapes based on the Uses within the View color fills the object shapes based on the value of the age characteristic for that object color fills the object shapes based on the value fo the technical risk ranking characteristic of that object color fills the object shapes based on the value of operational risk characteristic of the object or its associated control

64. “The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated

characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” *Id.*, 5:27-44.

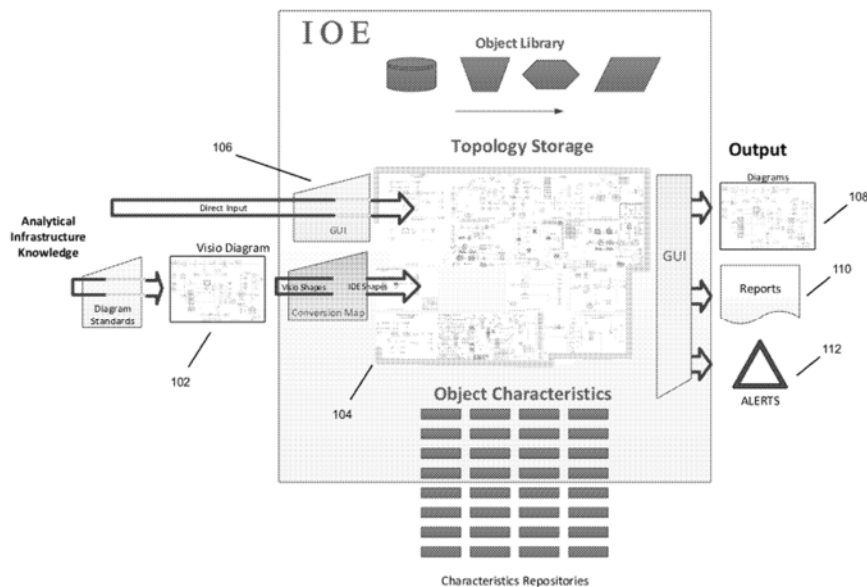


FIGURE 1

Id., Fig. 1.

65. As shown in, *e.g.*, Fig. 1, “[a] map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the

IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” *Id.*, 18:36-50.

2. Zheng (EX-1005)

66. U.S. Patent No. 10,503,907 (“Zheng”), entitled, “Intelligent Threat Modeling and Visualization,” was filed on December 14, 2016, published on June 15, 2017, and claims priority to December 14, 2015. I am informed that Zheng is prior art to the ’366 under 35 U.S.C. § 102(a)(2).

67. Zheng is titled “Intelligent Threat Modeling and Visualization” and discloses “[a] computer-implemented system... for visualizing and analyzing security threats in a suite of software applications.” EX-1005, Abstract. “The system includes a visualization module for rendering, on a computer display, a map with components representative of the suite of software applications and

relationships among the software applications.” *Id.* “The threat modeling module is adapted to interact with the visualization module to graphically depict on the computer display the one or more security threats on the map in a threat modeling layer.” *Id.*

68. Within the threat modeling layer, Zheng discloses that “[t]o generate and render the different layers of the multilayered structure 200, the visualization module 116 can receive and process data from a variety of sources, as illustrated in FIG. 1.” *Id.*, 4:28-31.

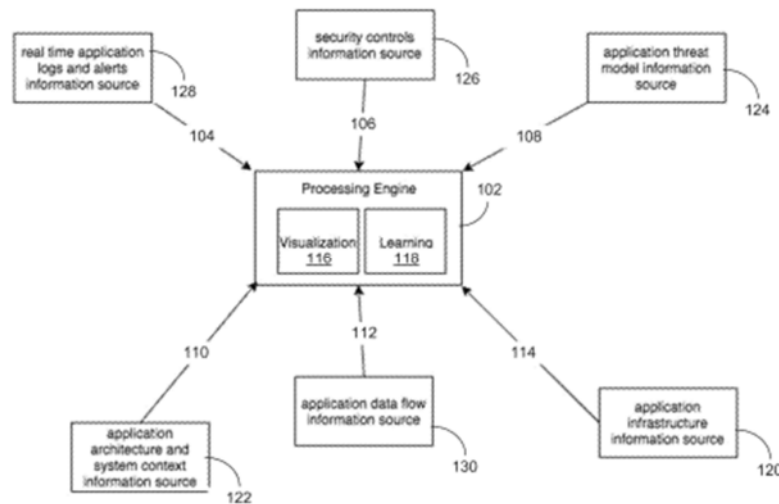


FIG. 1

69. Zheng states, “As shown, the visualization module 116 is adapted to receive (i) at input 114 information 120 related to the infrastructure of applications in one or more suites of applications and (ii) at input 110 information 122 related to the architecture of the individual applications and their system context. Such data 120, 122 can describe the complexity of one or more application suites with

many dependencies and connections.” *Id.*, 4:44-51; *see also* 9:48-53 (“When a new component (e.g., a new software application) 606 is detected in the application ecosystem, either manually or using an automated data gathering approach (e.g., scanning), the new component 606 can be added to the base layer 202 as a new feature vector 608.”).

3. Baker (EX-1006)

70. U.S. Patent Pub. No. 2014/0236665 (“Baker”) was published no later than August 21, 2014. I am informed that Baker is prior art under 35 U.S.C. § 102(a)(1).

71. Baker discloses systems related to the identification of risks and implementing procedures for managing those risks. EX-1006, ¶[0002]. The systems allow a user to create risk management process flowcharts. *Id.*, ¶[0025]. Additionally, “risk management or control flowcharts may be imported at step 104 from third party software such as that marketed as Microsoft Office Visio®,” as shown in Fig. 4:

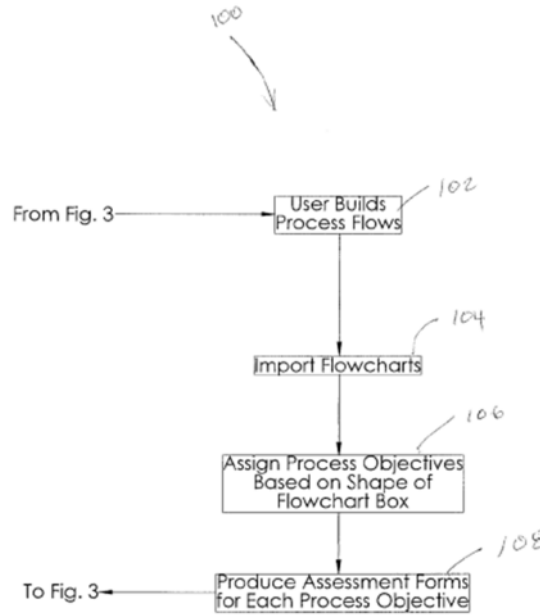


Fig. 4

Id.

72. “The process 100 then proceeds to step 106. The text and shape of each element of the flowchart are stored in the process database 30 in memory 26. At step 106, the processor 20 acts to automatically assign or associate predetermined objectives or controls according to the shapes of the imported flowcharts. It is understood that the predetermined data may have been stored previously in the process database 30 of the memory 26. This process data may later be retrieved for use in initiating process searches over the internet 12 as will be described later. The process 100 then proceeds to step 108 at which assessment forms or other documentation may be generated or stored for each process

objective. Process is then received to step 52 of process 50 illustrated in FIG. 3.”

Id.

4. Jones (EX-1007)

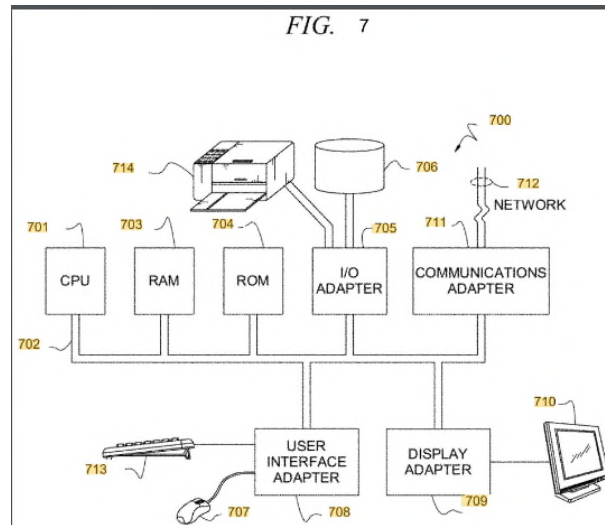
73. U.S. Patent No. 9,602,529 (“Jones”) was filed on April 2, 2014, published on October 8, 2015, and issued on March 21, 2017. I am informed that Jones is prior art under 35 U.S.C. § 102(a)(1).

74. Jones discloses a system for generating threat models. EX-1007, Abstract. The system includes “reporter 114” that generates reports regarding the threat model. EX-1007, Figure 1.

5. Galliano (EX-1008)

75. U.S. Patent No. 10,681,068 (“Galliano”) was filed on January 4, 2017, published on June 9, 2020, and claims priority to an application filed July 26, 2016. I am informed that Galliano is prior art under 35 U.S.C. § 102(a)(2).

76. Galliano discloses a “system and method for joining various cyber security threat data sources into a single threat data model, analyzing this threat data and storing the data as a specific data model...” Abstract.



B. Ground 1: Keenan Anticipates the Challenged Claims

77. In my opinion, Keenan anticipated the Challenged Claims. All of the limitations of the Challenged Claims are disclosed in Keenan. My analysis comparing Keenan to the Challenged Claims is contained below. I also have included additional analysis and citations to Keenan in the chart attached as **Exhibit B.**

1. Claim 1

a. Element 1[p]. “A threat modeling method, comprising:”

78. In my opinion, Keenan discloses this preamble. Keenan discloses an Integrated Object Environment (“IOE”) with a Graphical User Interface (“GUI”), where “[w]ithin a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box

creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View.” EX-1004, 18:6-13; *see also, e.g.*, Abstract, 4:64-67, 7:41-59, 13:4-22, 14:35-39, 14:61-63 (“Technical Threat Vector”), 16:1-29, 16:40-5, 17:32-40, 18:6-13, 20:34-51, Figs. 7, 16.

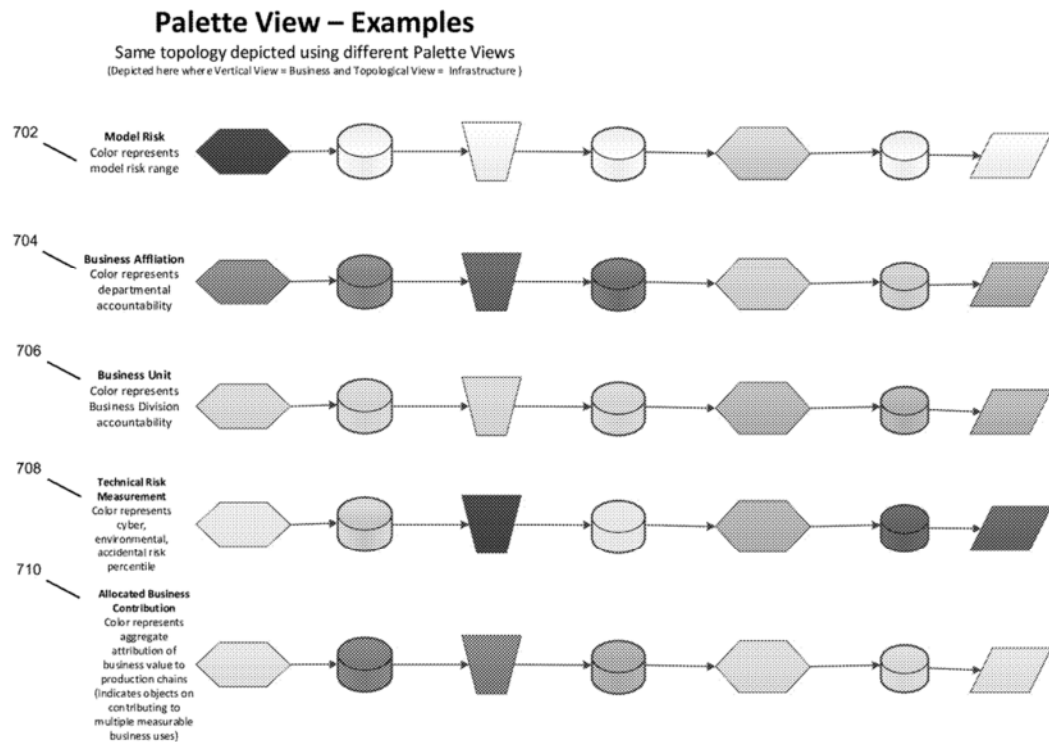


FIGURE 7

As shown in Fig. 16, a user may process and view objects in a variety of ways, called “Views,” such as by threat vectors, which “fills the object shapes based on the value of the technical risk ranking characteristics of that object,” *id.*, 16:21-23,

14:7-15:10, Fig. 16. These Views may be combined. *Id.*, 14:7-9. Thus, for example, the View for “model risk” may be combined with “threat vector.” *Id.*

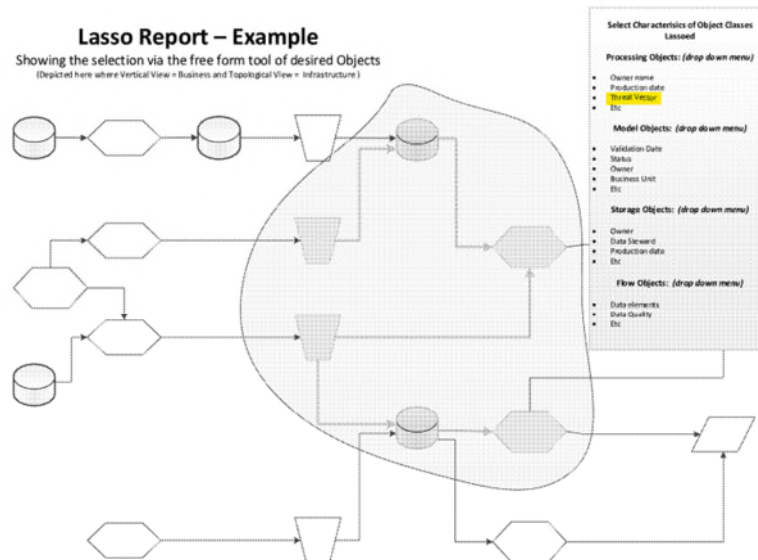


FIGURE 16

79. A POSITA would understand that Keenan discloses a threat modeling system and method. Keenan discloses “an Integrated Object Environment (IOE) running in a graph database environment,” which associates “risk” (including threats) to assets (*i.e.*, sources of threats / threat model components). *Id.*, Abstract; 1:31-46. Keenan’s method and system includes risk analysis, such as “storing, revealing and maintaining ... risk of information assets.” *Id.*, Abstract. A POSITA would understand that a determination of threats is included within a risk analysis. Keenan further explains that certain Views use color to “represent different types of risk (e.g., cyber, environmental, accidental, etc.).” *Id.*, 16:40-51. A POSITA would further understand that “risk” as disclosed in Keenan includes

threats because measuring risk includes a determination of threats. Keenan demonstrates that its risk measurement and display includes a determination of threats where Keenan discloses a risk measurement view that uses use color to “represent different types of risk (e.g., cyber, environmental, accidental, etc.).” *Id.*, 16:40-51. A POSITA would understand that “types of risk,” as disclosed in Keenan are threats because the described examples are all types of threats (cyber, environmental, accidental). Thus, Keenan discloses threats within its risk analysis. Moreover, a POSITA would understand that risk is a function of threats exploiting vulnerabilities to obtain, damage, or destroy assets. In other words, threat modeling is used to identify risk. The prior art references cited herein demonstrate the connection between risk analysis and threat modeling. *See, e.g.*, EX-1005, 1:37-45 (“Accordingly, there is a need for improved systems, methods and apparatuses for a more effective and efficient threat modeling approach in the context of application architecture and infrastructure. For example, the present invention features a system that can dynamically produce interactive and predictive threat models that can support real-time (or near real-time) risk analytics based on automated data aggregation and transformation, easy-to-understand visualization and sophisticated machine-learn techniques.”), 5:25-35 (“These security risks can be manually and/or automatically identified by a threat modeling module 124 that is configured to interact with the visualization module 116 and supply data to the

visualization module 116 via the input 108 for graphically depicting the identified risks. In one example, identification of a security threat can be performed manually by security personnel.”); EX-1007, 1:52-58 (“Example implementations of the present disclosure may facilitate an understanding of these information systems, and cybersecurity threats to which they may be susceptible. This understanding may facilitate the design or modification of an information system to mitigate or further mitigate the risk of those threats, to an acceptable if not desired level.”); EX-1011 (excerpts of treatise), p. 94 (“There are several good reasons to integrate threat modeling activities within the SDLC (software development lifecycle). Threat models help information security reviews of software projects and assess the risks posed by vulnerabilities identified during the various type security activities that are built into SDLC such as design reviews, secure coding, and penetration tests. A threat model help risk managers to assess application security risks by correlating the analysis of threats to the vulnerabilities to application data and functional assets. A risk analysis that factors threats, vulnerabilities, and assets can also factor likelihoods and impacts to determine the risk level and to apply security measures to reduce this level of risk.”). Indeed, the ’366 patent itself recognizes that threat modeling is used for risk analysis. EX-1001, 2:3-20 (“traditional threat modeling systems and methods exist for assessing risk from potential cyber threats...”), 2:45-46 (discussing “risk from potential threats.”),

14:5-17 (discussing a threat’s “risk level”), 15:26-45, 15:64-16:3, 17:60-64 (noting that threat reports include a “threats” tab that lists all of the threats, “identifying the risk level of each threat”), 18:9-12, 22:13-27, 22:28-37, 23:7-12, 23:42-48, 24:53-61, 25:7-12, 25:20-29, 26:38-45 (listing “composition of threats by risk”), 26:50-58, 32:59-62, 34:31-39, 35:35 (“THREATS BY RISK area” on user interface), 35:50 (“an overall threat risk of the model”), 39:47-54. Keenan discloses methods and systems of measuring risk through threat modeling by providing “visual depictions” and models of the risk (which includes threats), *i.e.*, through visually depicting threats against IOE objects and creating “Views” through threat vectors, which may be combined with other Views, such as “model risk.” *Id.*, 14:7-15:10. For example, Keenan’s system provides a method to “calculate the risk of the selected View,” which creates a “visual depiction” of that risk and giving the user an option to print “risk reports,” which can include the aggregate model risk for all models in the view. *Id.*, 18:6-14. Keenan further describes associating “objects,” which include various assets/sources of threats, with characteristics (structural metadata), which may include a risk measurement. *Id.*, 4:64-67; *see also* 9:42-47 (noting that an example name of a threat model could be “European Windstorm Model”), 14:7-15:10, Figs. 1, 7, 16. Based on these disclosures, a POSITA would understand that Keenan discloses a threat modeling system and method.

b. Element 1[a]. “providing one or more databases, the one or more databases comprising:”

80. In my opinion, Keenan discloses this element. Keenan discloses a system and method that is run in a “graph database environment,” which is used for “storing, revealing and maintaining value and risk of information assets.” EX-1004, Abstract (“A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.”). Moreover, Keenan discloses a system and method that converts Visio files “into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” *Id.*, 5:37-44; *see also* 5:45-62, Figs. 1, 4.

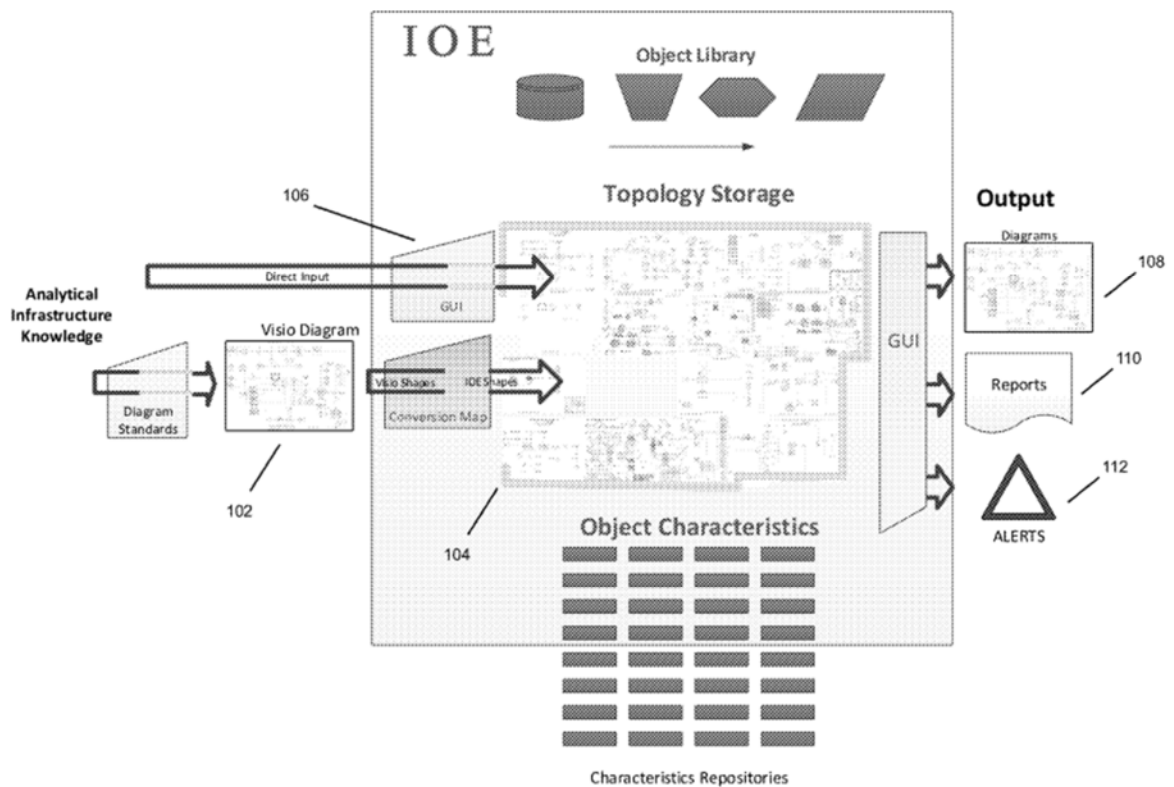


FIGURE 1

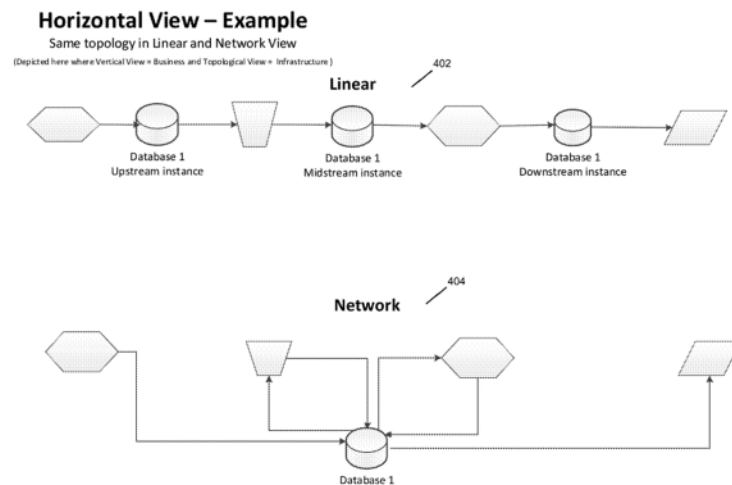


FIGURE 4

- c. **Element 1[b]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is**

associated with at least one of the threat model components through the one or more databases;”

81. In my opinion, Keenan teaches this element. As discussed above, the PO defined “threat model components” as “relevant sources of threats.” See Section VII.C. Keenan discloses “a plurality of threat model components” and “a plurality of threats” “wherein each of the threats is associated with at least one of the threat model components” because Keenan teaches an IOE that represents “a dynamic map of an enterprise’s analytic infrastructure, preferably including all physical assets that contribute to the analysis.” EX-1004, 5:27-44. Each asset or “real world component,” *i.e.*, physical asset, is defined by an object within the IOE and is associated with risk, as well as characteristics such as the relationship to other objects. *Id.*; *see also id.*, 1:33-38. One such characteristic is associated risk or “risk characteristics.” The IOE allows a user to select views that display the associated risk and consequences for specific assets using threat vectors and models. *Id.*, 4:64-67, 16:1-29 (“Palette View options allow the user to highlight a particular set of Characteristics that are preset such as risk”); *see also* 1:31-52 (the IOE “associates the risk ... of the asset to the product chain ... [and measures the] consequences of using those assets”), 7:41-59, 13:4-22, 14:35-39, 14:61-63, 16:40-63, 17:32-40, 18:6-22, 20:34-51 (“calculate risk parameters based on new raw values put into to characteristics”), Figs. 1, 7. Keenan further explains that certain

Views use color to “represent different types of risk (e.g., cyber, environmental, accidental, etc.).” *Id.*, 16:40-51.

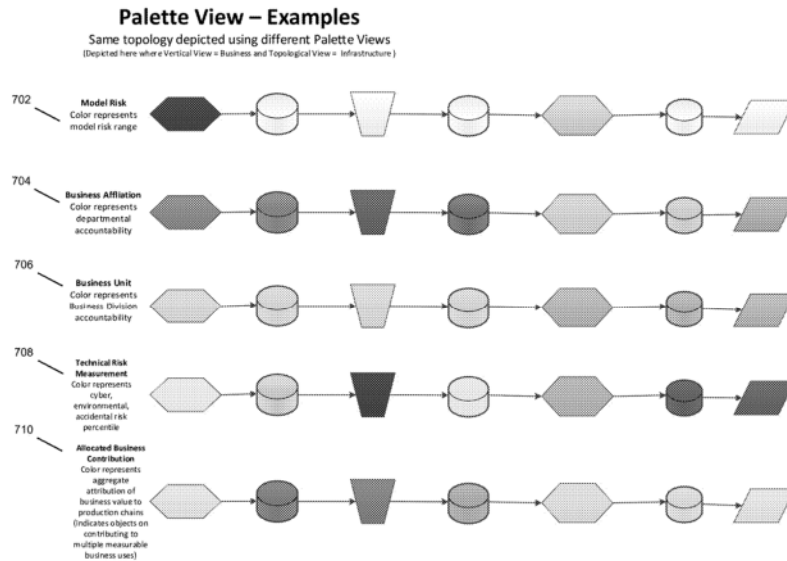


FIGURE 7

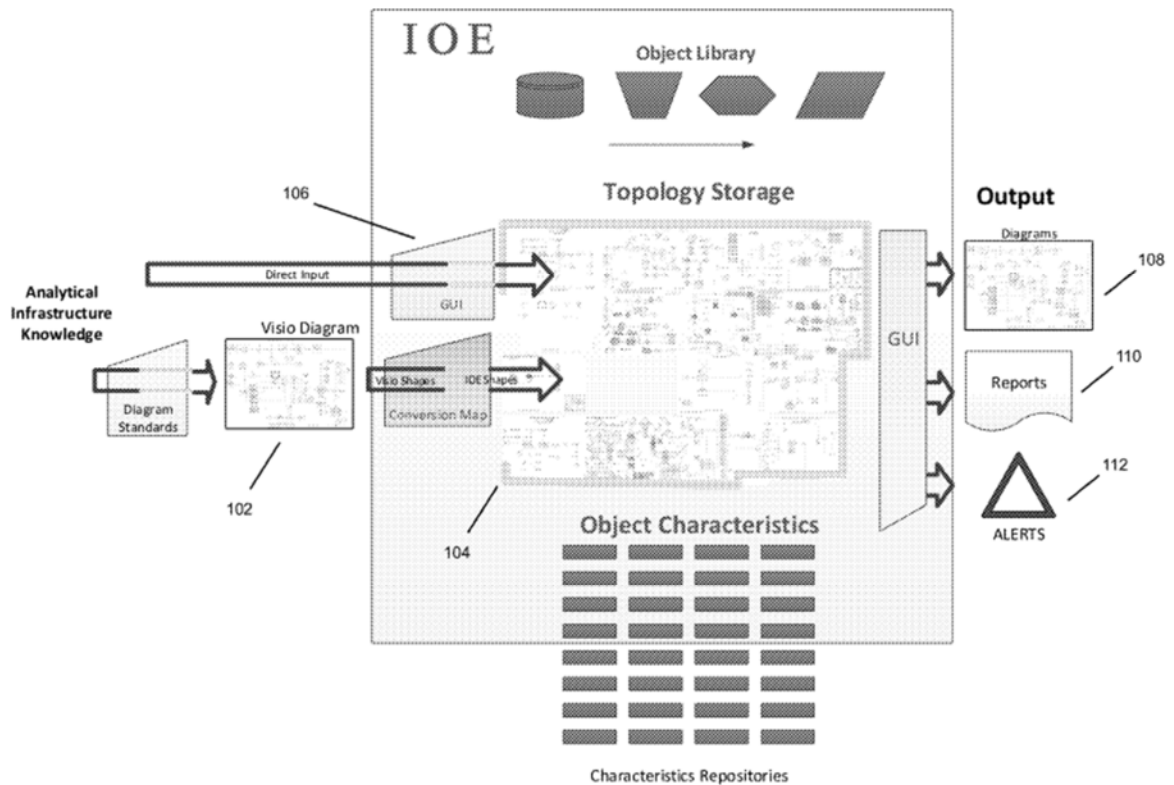


FIGURE 1

82. A POSITA would understand that the assets/objects in Keenan are “relevant sources of threats” and thus threat model components because the assets/objects in Keenan are subject to risks that Keenan’s system measures. A POSITA would further understand that “risk” as disclosed in Keenan includes threats because measuring risk includes a determination of threats. Keenan demonstrates that its risk measurement and display includes a determination of threats where Keenan discloses a risk measurement view that uses color to “represent different types of risk (e.g., cyber, environmental, accidental, etc.).” *Id.*,

16:40-51. A POSITA would understand that “types of risk,” as disclosed in Keenan are threats because the described examples are all types of threats (cyber, environmental, accidental). Thus, Keenan discloses threats within its risk analysis. A POSITA would further understand that “each of the threats is associated with at least one of the threat model components” because Keenan discloses associating risk with each asset. *Id.*, 1:33-38. Keenan further provides a system and method to “calculate the risk of the selected View,” which creates a “visual depiction” of that risk and giving the user an option to print “risk reports.” *Id.*, 18:6-13. Keenan further describes associating “objects,” which include various assets and sources of threats, with characteristics (structural metadata), including “risk characteristics,” which may include a risk measurement. *Id.*, 4:64-67. These characteristics include, among other things, the relationship between objects. *Id.*, 5:27-44. Keenan further describes the consequences of using an asset (component). *Id.*, 1:47-52; *see also* Element 1[p].

83. A POSITA would further understand that Keenan’s threats and threat model components are “stored therein” in the database, and that the threats and threat model components are associated “through the one or more databases” because Keenan discloses a method and system that are run in a “graph database environment,” which is used for “storing, revealing and maintaining value and risk of information assets.” *See* Element 1[a].

d. Element 1[c]. “providing one or more mapping files communicatively coupled with the one or more databases,”

84. In my opinion, Keenan discloses this element. Keenan teaches that “[a] map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram.” EX-1004, 18:36-40. Keenan further discloses a GUI with a variety of views, including a Palette View, that shows model risk. *Id.*, 16:5-11, Fig. 7; *see also* 5:27-44 (“The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise’s analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file.”), 11:65-12:4, 12:11-62, 16:5-11, 18:23-28

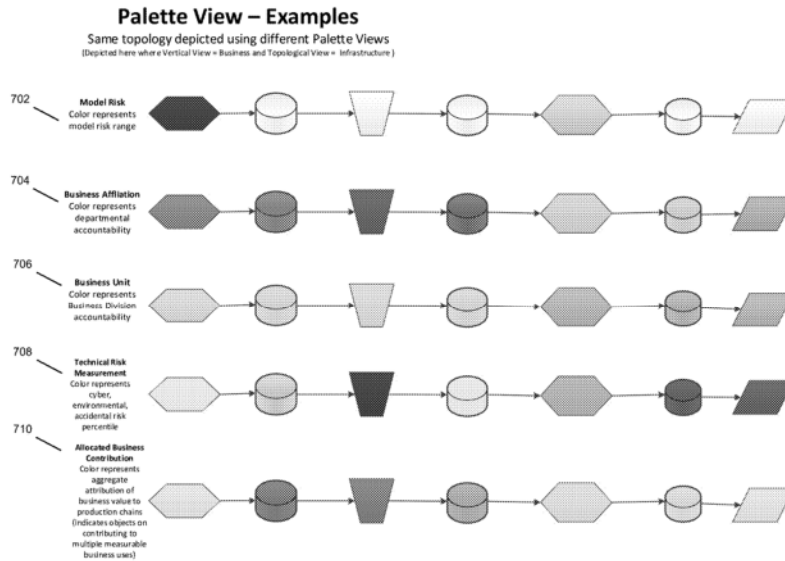


FIGURE 7

85. A POSITA would understand that Keenan’s disclosure of a “map” discloses a “mapping file” because a mapping file is required to form a map. In other words, Keenan’s “map” conveys, and necessarily includes, a “mapping file” as claimed because a mapping file would be required to perform mapping techniques.

86. A POSITA would further understand that Keenan’s mapping file is “communicatively coupled with the one or more databases” because Keenan discloses software that “converts the graph database and associated characteristics into Views.” *See* Element 1[a]. In other words, the data stored in Keenan’s database, such as the assets and associated characteristics, are turned into Views, *i.e.*, maps, by the mapping file—demonstrating a communicative coupling between the mapping file and the graph database.

- e. **Element 1[d]. “the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”); and”**

87. In my opinion, Keenan discloses this element. Keenan teaches that “[a] map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match.” *Id.*, 18:36-40. As discussed in Element 1[b], the IOE objects/assets are threat model components. *See also* Element 1[p]. Thus, Keenan discloses “mapping,” *i.e.*, through one or more mapping files, Keenan’s threat model components with Visio shapes and formats. A POSITA would understand that the Visio shapes and formats in Keenan are “visual diagram components” because a user the Visio shapes and formats are “mapped,” *i.e.*, visually associated, with objects, which Keenan’s system visualizes, and because “shapes and formats” are tools used for visualization. *Id.*, 18:6-40 (“The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function. Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode

is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.”); *see also id.*, 5:27-44, 18:23-50, 19:17-29, 21:10-17, Figs. 12-13.

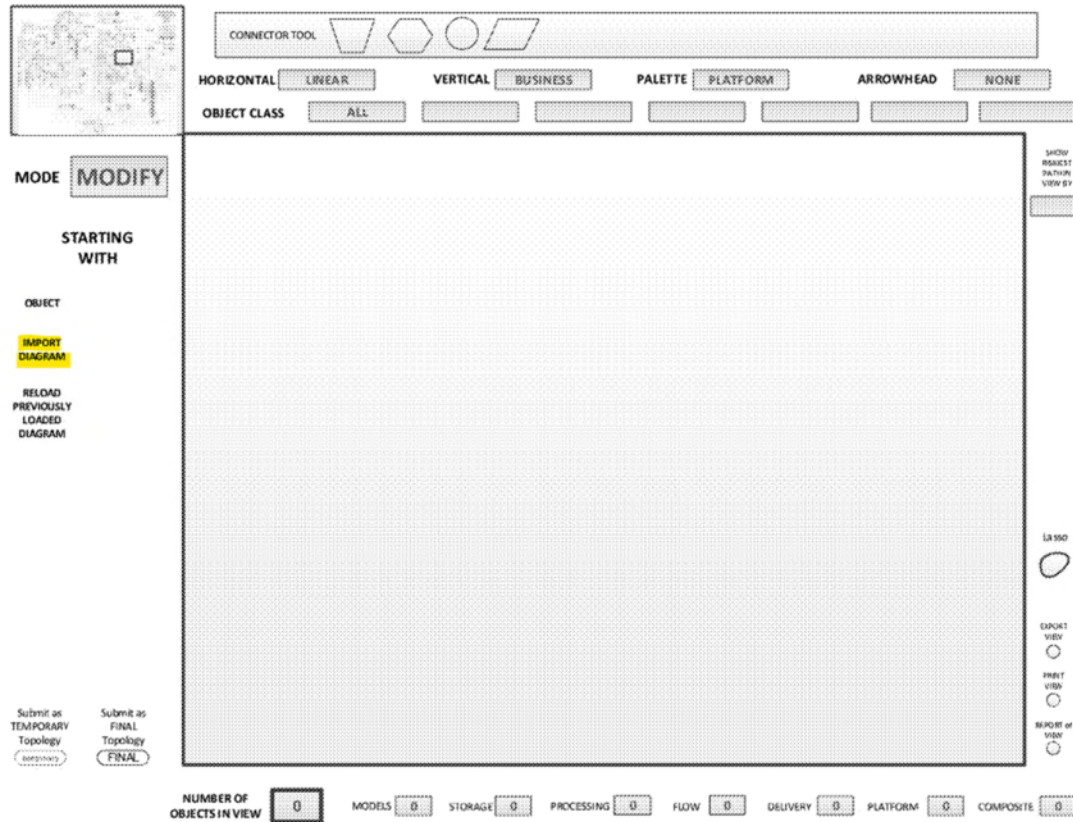


FIGURE 12

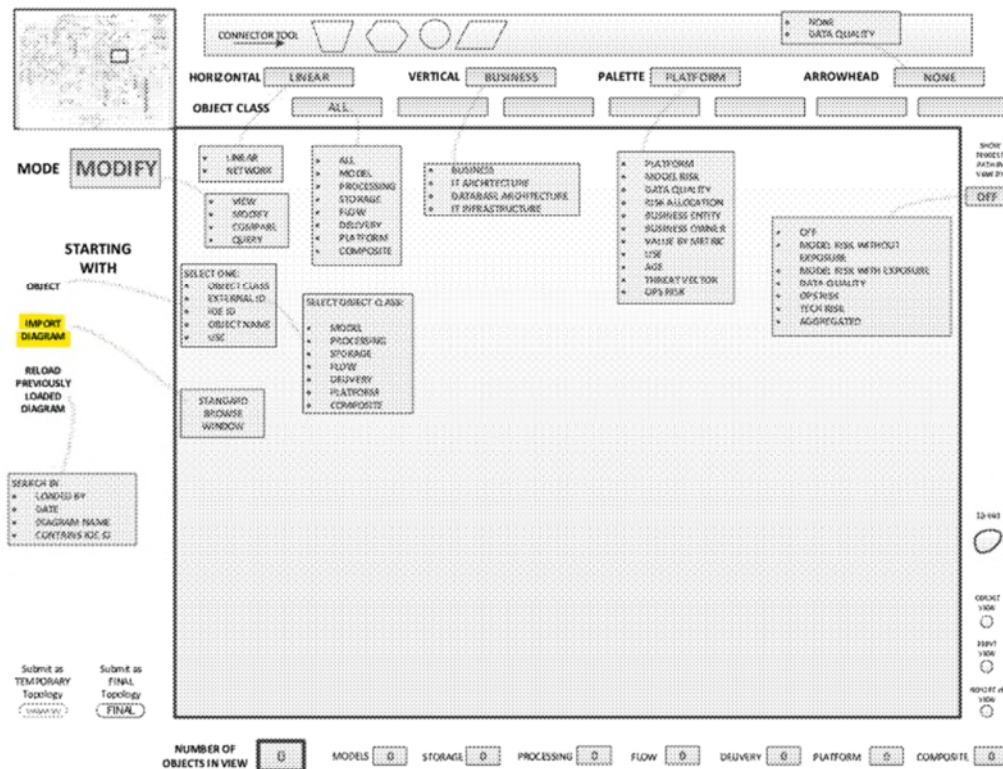


FIGURE 13

88. A POSITA would further understand that the Visio shapes and formats are “of a third party software application” because user of Keenan’s system selects the option to “import” the Visio visual diagram components to Keenan’s system. *Id.*, 18:36-40. Further, Keenan teaches a system and method that “converts” Visio files “into a standard, canonical format for storage.” *Id.*, 5:37-44. Moreover, as discussed, PO defined a “third party software application” as “independently-created software applications not created with a common schema, such that a mapping file or the like as claimed in the claims would be useful to map between elements of the two software applications.” A POSITA

viewing Keenan would understand that Visio is an independently-created software application not created with a common schema because Keenan specifically employs a mapping file to map between the elements of Visio and Keenan's system. *Id.*, 18:36-40.

f. Element 1[e]. “in response to receiving one or more user inputs, using one or more user interfaces displayed on one or more computing devices communicatively coupled with the one or more databases:”

89. In my opinion, Keenan discloses this element. Keenan teaches “in response to receiving one or more user inputs, using one or more user interfaces displayed on one or more computing devices” because Keenan discloses that “[i]n embodiments of the present system, there are preferably four modes for the user to interact with the IOE via a Graphical User Interface (GUI).” EX-1004 at 11:42-43, 18:6-11. Figs. 2 and 3 show “an exemplary GUI”:

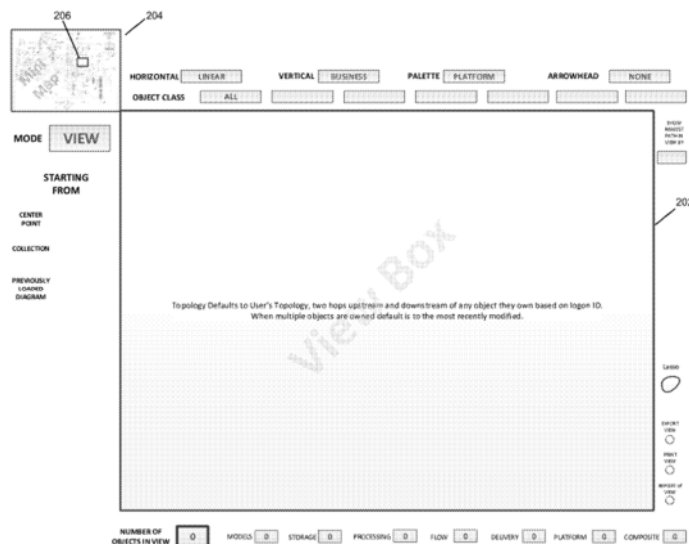


FIGURE 2

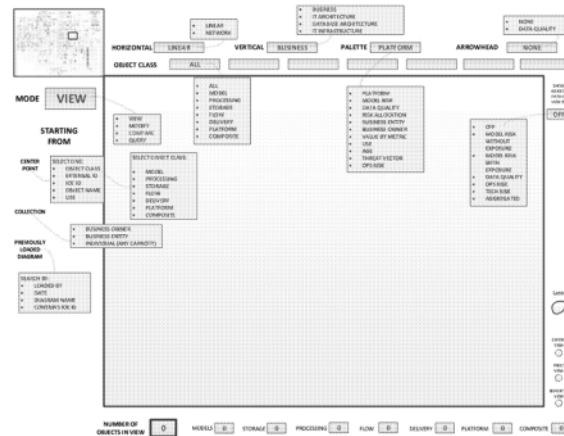


FIGURE 3

See also, e.g., *id.*, Abstract, 1:53-60, 2:3-6, 4:5-14, 5:24-27, 5:40-44, 11:42-64, 12:64-14:10, 18:36-50, Figs. 4-16.

90. Keenan further teaches “one or more user interfaces displayed on one or more computing devices” because Keenan discloses that its systems and method is “implemented by relevant computing hardware instructed by a program.” *Id.*, 21:26-33.

91. A POSITA would further understand that the user interface and the computing device are “communicatively coupled with the one or more databases” because Keenan discloses that the GUI and computing device are used to interact with the IOE—and the IOE runs on a graph database which is then implemented by the computing device. See *id.*; Element 1[a].

- g. Element 1[f]. “using the one or more computing devices, reading a data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

92. In my opinion, Keenan discloses this element. Keenan teaches “reading a data file generated by the third party software application” because Keenan discloses that a “map is preset that *maps Visio shapes and formats to IOE shapes and formats.*” EX-1004, 18:36-40; Element 1[d]. A POSITA would understand that that the Visio shapes and formats imported to Keenan’s IOE are “a data file generated by a third party software application.” See Element 1[d]. A POSITA would further understand that the data file must be “read,” for example, when imported and when Keenan’s system “converts the Visio files 102 into a standard, canonical format.” EX-1004, 5:27-44.

93. Keenan further discloses that “the data file comprising a subset of the third party diagram components.” See Element 1[d].

94. Keenan further discloses “the data file defining one or more relationships between the subset of third party diagram components” because Keenan discloses that “each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database” and defines the “one or more relationships between the subset of third party diagram components.” EX-1004, 5:33-40. Keenan’s system further includes “data flow objects” that represent

the relationships between objects. *Id.*, 1:24-30, 4:39-45, 4:54-60, 5:51-62, 7:25-34, 8:21-41, 15:15-43.

95. Keenan further discloses that the data file is read in response to user inputs because Keenan discloses that “[t]he user selects the import function in the IOE GUI” to bring in Visio shapes and formats. *Id.*, 18:36-40; Element 1[d].

Keenan further discloses that its system “converts the Visio files 102 into a standard, canonical format.” EX-1004, 5:33-40. A POSITA would understand that the data file (Visio file) must be read in response to user inputs because a computing device must read a data file to import or convert the data file after the user selects the import option of Keenan’s system.

96. A POSITA would understand that this element is performed “using one or more computing devices.” See Element 1[e].

h. Element 1[g]. “determining using the one or more computing devices, for the subset of third party diagram components, correlated threat model components as defined in the one or more mapping files;”

97. In my opinion, Keenan discloses this element. Keenan teaches “determining ... for the subset of third party diagram components, correlated threat model components as defined in the one or more mapping files” because Keenan discloses that a “map is preset that maps Visio shapes and formats to IOE shapes and formats. ... The existing Objects are then updated with new relationships

based on the imported Visio diagram.” EX-1004, 18:36-50; *see also id.*, 5:27-44, 18:23-50, 19:17-29, 21:10-17; Element 1[c]-1[d].

98. A POSITA would understand that Keenan’s disclosure describes “determining” the claimed relationships of which threat model components are correlated with the third party diagram components in the mapping file because “existing objects are updated with new relationships” based on the importation of Visio diagrams—*i.e.*, the system must determine the relationships before updating new relationships.

99. Keenan further discloses this limitation is performed in response to user inputs because Keenan discloses that “[t]he user selects the import function in the IOE GUI” to bring in Visio shapes and formats. *Id.*, 18:36-40. Keenan’s system then determines, for the subset of third party diagram components (*i.e.*, Visio shapes and formats) correlated threat model components (*i.e.*, assets/objects). *See* Elements 1[b], 1[d], 1[e].

100. A POSITA would understand that this element is performed “using one or more computing devices.” *See* Element 1[e].

- i. **Element 1[h]. “displaying on the one or more user interfaces a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

101. In my opinion, Keenan discloses this element. Keenan teaches “a relational diagram” because Keenan discloses an IOE including “[d]iagrams of the production chain of information assets. ... These diagrams depict data flows linking together these information assets.” EX-1004, 1:24-30. These diagrams are displayed on the GUI and “serve[] as a communication medium to and from users and the IOE.” Id., 2:5-6. The diagrams show the “[r]epresentation of the relationships between objects in an analytical infrastructure,” id., 5:5-10, and are thus “relational diagrams” as claimed. A POSITA would understand that the interface displays a relational diagram that includes “one of a system, an application, and a process” because Keenan discloses these relationships in the relational diagram “represent the storage, processing and flow of information,” id., 5:5-10, as well as other diagrams or “Views” (which Keenan notes is synonymous with diagrams, id., 5:24-27) showing systems, applications, and processes, id., 1:24-33, 2:5-17, 4:5-14, 4:46-60, 5:5-10, 5:17-19 (topology “[d]epicts tasks in process order sequence”); 7:60-62 (“Platform Objects include physical and/or software systems that house...models”), 9:56-63, 10:37-50, 11:42-64, 18:66-19:5 (“The icon [in the GUI] represents the document format. By clicking on this icon, the application which supports the document opens and the document appears on the screen.”), Figs. 1-3, 9, 12-15. Moreover, a POSITA would understand that the IOE GUI display as a whole with the imported Visio diagram components

represents, as shown in the Figures, *e.g.*, Figs. 1 and 10, are “one of a system, an application, and a process.”

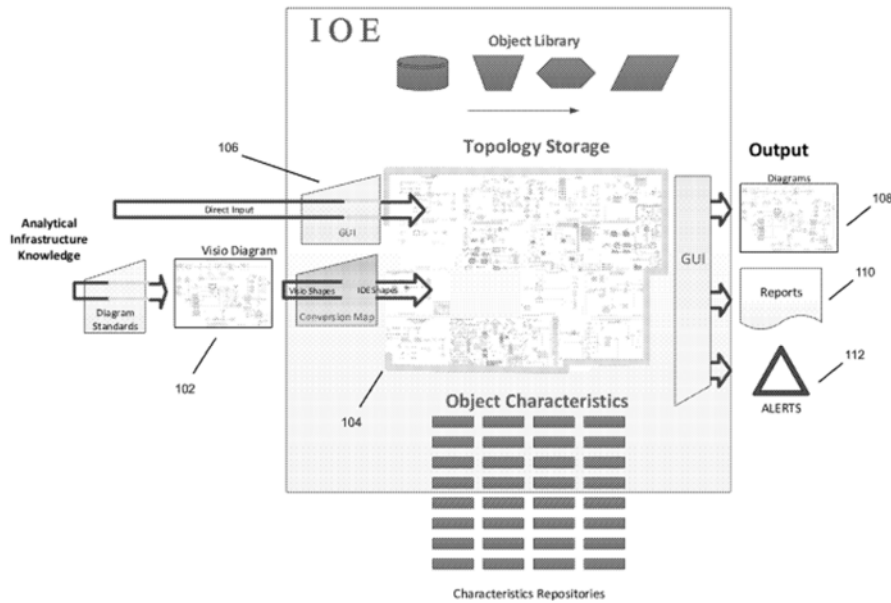


FIGURE 1

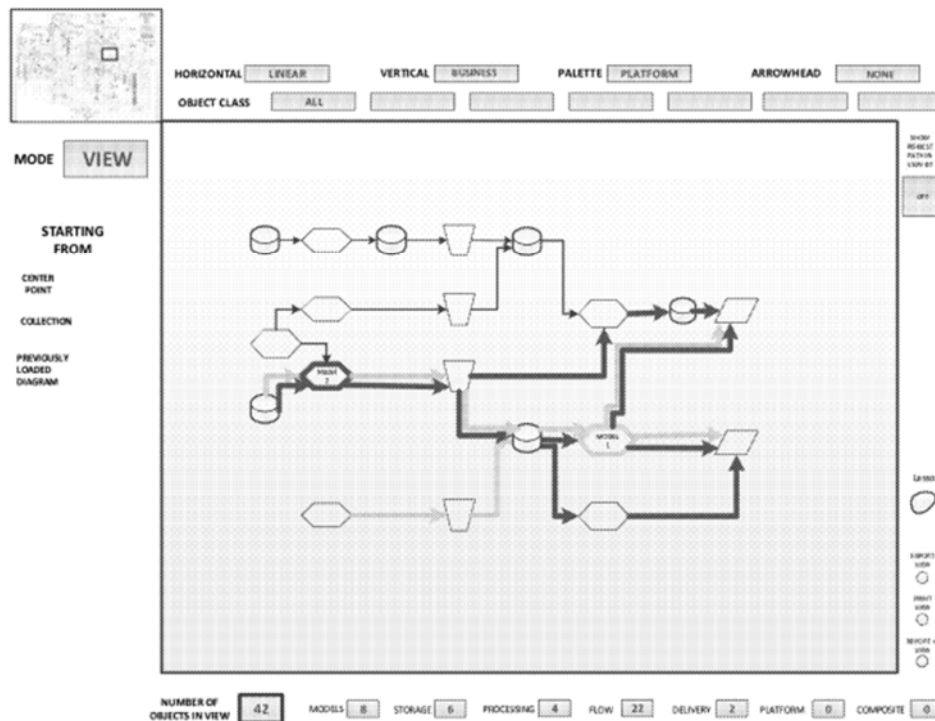


FIGURE 10

102. A POSITA would further understand that this relational diagram of one of a system, an application, and a process is displayed on the one or more interfaces “using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model” because Keenan discloses that “existing Objects are [] updated with new relationships based on the imported Visio diagram”—i.e., imported Visio diagrams associated with objects, see Element 1[d], are updated to “represent the storage, processing and flow of information,” and/or other systems and processes cited above.

103. Keenan further discloses “the relational diagram defining a threat model.” *See* Element 1[p].

104. Keenan further discloses this limitation is performed in response to user inputs because Keenan discloses that “[t]he user selects the import function in the IOE GUI” to bring in Visio shapes and formats. *Id.*, 18:36-40. Keenan’s system then determines, for the subset of third party diagram components (*i.e.*, Visio shapes and formats) correlated threat model components (*i.e.*, assets/objects) and associated visual representations. *See* Elements 1[b], 1[d], 1[e].

105. Keenan further discloses that the display is on “one or more user interfaces.” *See* Element 1[e]. Moreover, Keenan discloses that “[d]iagrams are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams ... serve[] as a communication medium to and from users and the IOE.” EX-1004, 2:3-6.

- j. Element 1[i]. “generating, using the one or more computing devices, and displaying, on the one or more user interfaces, a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

106. In my opinion, Keenan discloses this element. Keenan teaches that “a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts

112.” EX-1004, 5:40-44; *see also id.*, *see also id.*, 8:15-20, 17:11-40, 18:6-22, 19:30-65, Figs. 1, 3, 5, 11-15, 16. Keenan’s report displays each threat because the characteristics, *i.e.*, “risk characteristics,” may be “converted into a Report,” as well as because Keenan allows for specialized “Lasso reports,” that allow for user-selected reports. *Id.*, 17:38-40, Fig. 16. As shown in Fig. 16, a user may select a portion of the View (such as that the selection appears in a “Lasso” shape) to print in a report.

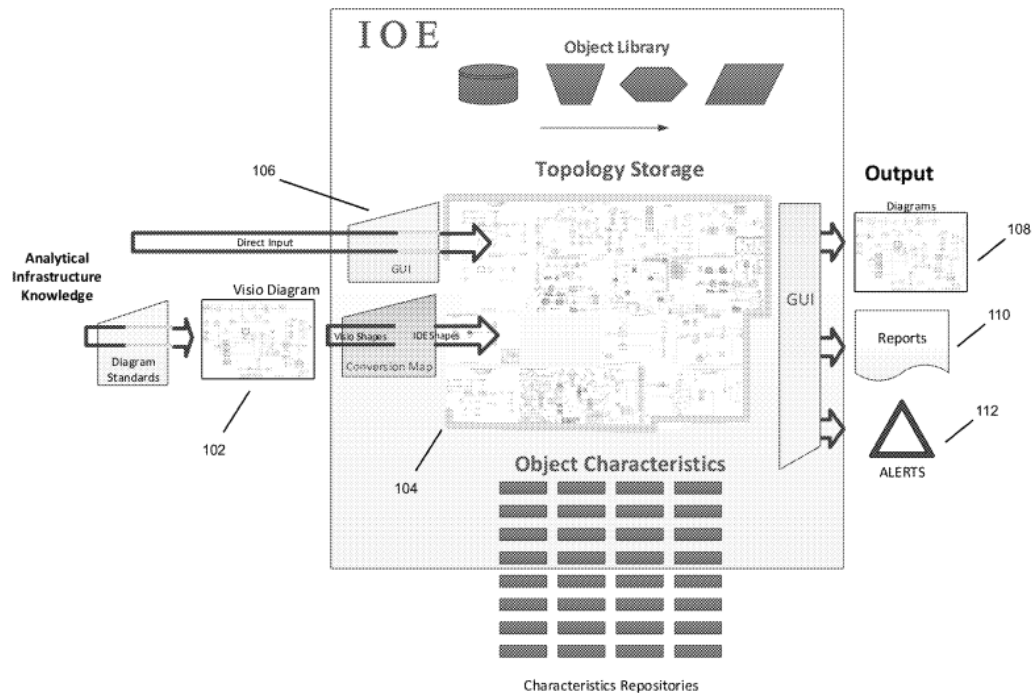


FIGURE 1

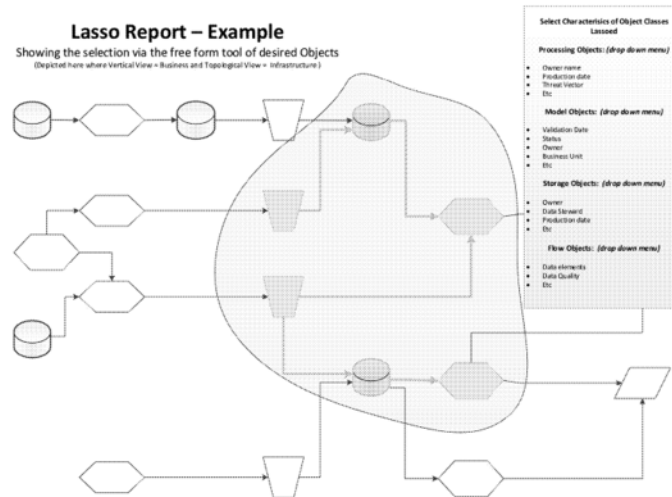


FIGURE 16

107. Keenan further discloses this limitation is performed in response to user inputs because Keenan provides software that its Views “are exportable as ... as reports.” *Id.*, 5:40-44. A POSITA would understand “exportable” to mean “the ability for a user to export” in context.

108. A POSITA would understand that this element is performed “using one or more computing devices.” *See* Element 1[e].

2. Claim 2: “The method of claim 1, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”

109. In my opinion, Keenan discloses this element. Keenan teaches that “[i]t is understandable to those skilled in the art that all or a part of steps of the processes in the preceding embodiments is preferably implemented by relevant computing hardware instructed by a program. The program may be stored in a

computer readable storage medium. The storage medium may include a ROM, a RAM, a magnetic disk or a compact disk.” EX-1004, 21:27-33. A POSITA would understand that Keenan discloses this element because Keenan discloses that the IOE is stored in a graph database, but further provides that programs implementing the IOE, such as the Visio mapping files, may be stored on a computer readable medium, including ROM or RAM. Computer readable storage mediums are mediums capable of storing data in a format readable by a mechanical device, and a POSITA would understand that Keenan’s reference to ROM or RAM indicates that files are stored on a computing device because ROM and RAM are both forms of computer storage.

3. Claim 3: The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.

110. In my opinion, Keenan discloses this element. Keenan teaches “[t]o connect objects on the screen, the GUI provides Flow Objects, which preferably connect to any point on the Object shape in the diagram to allow for clarity in the depiction of the topology. Double clicking on the object opens the characteristics table.” EX-1004, 18:65-19:2; *see also id.*, 8:16-20, 13:53-55.

4. **Claim 4: “The method of claim 1, wherein the data file comprises one of a VSD file, a VSDX file, a TMT file, an XML file, a JSON file, a PNG file, and a JPEG file.”**

111. In my opinion, Keenan discloses this element. As discussed, Keenan discloses a data file that comprises Visio shapes and formats. See Element 1[f]. A POSITA would understand that Visio files can be a “VSD file” or a “VSDX file.” For example, https://en.wikipedia.org/wiki/Microsoft_Visio states that Visio uses the VSD and VSDX file formats, which is consistent with my own knowledge and experience.

5. **Claim 5: “The method of claim 1, wherein determining, for the subset of third party diagram components, correlated threat model components, comprises first determining whether a user-defined name for one of the subset of third party diagram components has a match in the one or more mapping files and, if no match is found, then determining whether a default title for the third party diagram component, as defined by the third party software application, has a match in the one or more mapping files.”**

112. In my opinion, Keenan discloses this element. Keenan teaches that “[e]ach object preferably has two identifiers: one internal to the IOE, and one that ties the object to an authoritative external repository. ... Therefore, all ‘model’ objects in the IOE preferably carry their repository IDs as well as an ID internal to the IOE.” EX-1004, 6:13-26. Keenan further teaches that “[a] map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in

IOE based on external ID number first, then name match.” EX-1004, at 18:36-50.

A POSITA viewing Keenan would understand that an “external ID” is a default title as claimed and a “name match” or “Name of object” is or refers to a user-defined name.

113. Moreover, a POSITA would further understand that Keenan discloses a system that checks for default titles or user-defined name matches in either order because Keenan’s system allows a user to “select a method for adding an object, such as External Object ID ... and Name of object.” *Id.*, 18:51-60. The IOE then “returns a topology a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object.” *Id.* This is shown explicitly in Fig. 13:

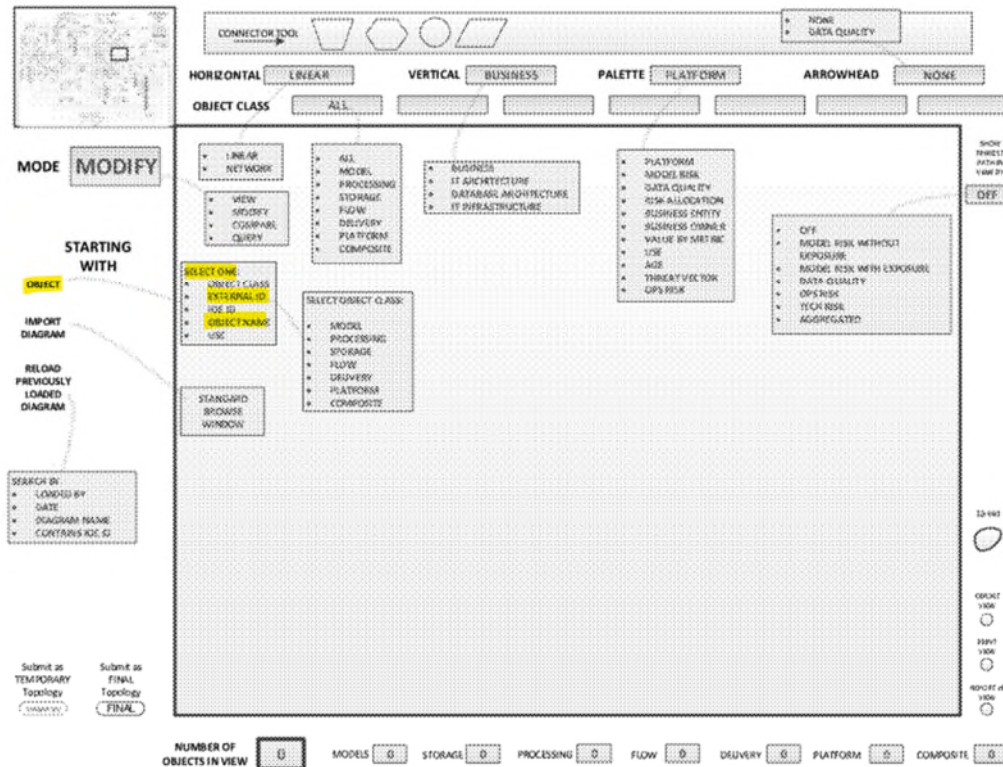


FIGURE 13

As the highlighted portions show, a user, starting with an object that the user wishes to add or import to the IOE, Keenan’s system may check for objects based on a variety of criteria, including “EXTERNAL ID” and “OBJECT NAME.”

6. **Claim 6: “The method of claim 5, further comprising, if no match is found for the default title, matching the third party diagram component with a placeholder component and displaying the placeholder component on the relational diagram to indicate that no match was found.”**

114. In my opinion, Keenan discloses this element. Keenan discloses that “[t]he user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following

objects matched existing objects from the IOE, new relationships will be loaded to existing object.” EX-1004, 18:36-50. If the object a user wants is not in the IOE, however, “the user can then add the object.” *Id.*, 18:51-60. A POSITA would understand that Keenan’s system creates a placeholder component where no exact match is found for the third party component object the user adds to Keenan’s IOE, and then displays that placeholder in the IOE, indicating no match was found. The user then may select the object to provide an “Object Name to identify an object,” so that the object is now part of the IOE and no longer a placeholder. *Id.*, 13:53-65.

115. Moreover, Keenan discloses the option to submit “temporary” topology objects in memory that are displayed on the GUI, as shown in Figs. 12 and 13:

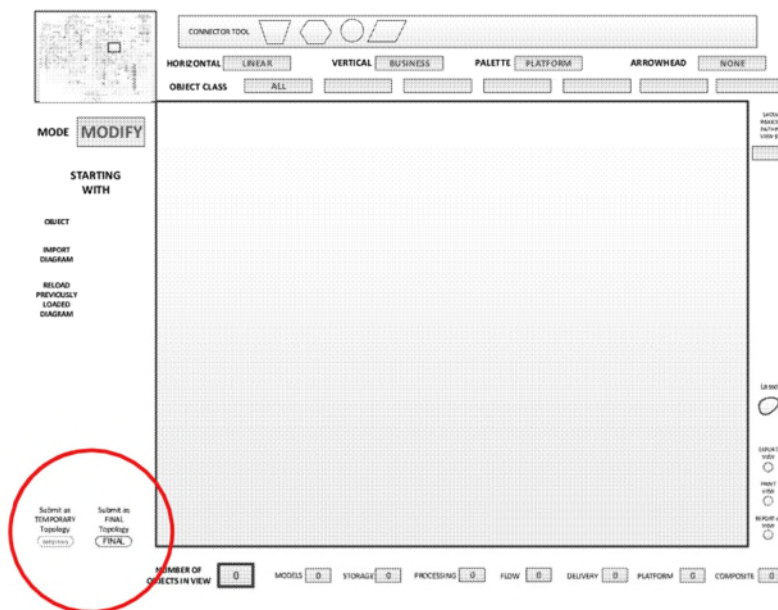


FIGURE 12

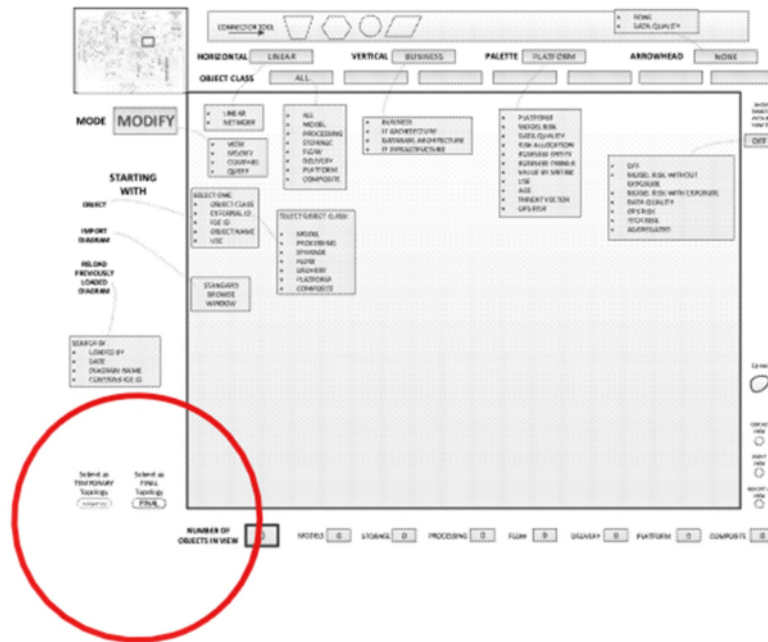


FIGURE 13

See also EX-1004, 11:7-11, 19:9-16.

116. A POSITA would understand that these temporary placeholder components are used where no match is found for the default title, and Keenan's system may match the third party diagram component Visio shapes and formats with a "temporary" placeholder component object and display that placeholder component on the relational diagram to indicate that no match was found.

7. Claim 7: "The method of claim 1, further comprising determining, for the one or more relationships defined by the data file, correlated threat model components as defined in the one or more mapping files."

117. In my opinion, Keenan discloses this element. As discussed in Element 1[f], Keenan discloses "the data file defining one or more relationships between the subset of third party diagram components" because Keenan discloses

that “each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database.” EX-1004, 5:33-40. Keenan’s system further includes “data flow objects” that represent the relationships between objects. *Id.*, 1:24-30, 4:39-45, 4:54-60, 5:51-62, 7:25-34, 8:21-41, 15:15-43. A POSITA would understand that these data flow objects represent, among other things, the one or more relationships defined by the data file.

118. Keenan’s system further provides “a method by which data flow objects could be separately linked to the components.” A POSITA would understand this disclosure to mean that the data flow objects, *i.e.*, the one or more relationships defined by the data file, may be correlated with the threat model components, *i.e.*, assets/objects, independently.

8. Claim 8:

- a. Element 8[p]. “A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:”**

119. In my opinion, Keenan discloses this element. A POSITA would understand that Keenan’s one or more databases is communicatively coupled with one or more computing devices because Keenan discloses that its system and

method is “implemented by relevant computing hardware instructed by a program.” *Id.*, 21:26-33; Elements 1[p]-1[a] and 1[e].

- b. Element 8[a]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;”**

120. In my opinion, Keenan discloses this element. See Elements 1[b].

- c. Element 8[b]. “one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);”**

121. In my opinion, Keenan discloses this element. See Elements 1[c]-1[d].

- d. Element 8[c]. “an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by one of the one or more computing devices, the data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

122. In my opinion, Keenan discloses this element. See Elements 1[e]-[f].

As shown in Figs. 12-13, Keenan discloses “an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by one of the one or

more computing devices” because Keenan discloses that “[t]he user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram.” EX-1004, 18:36-50; *see also* 18:6-35; 5:27-44, 18:23-50, 19:17-29, 21:10-17, Figs. 12-13.

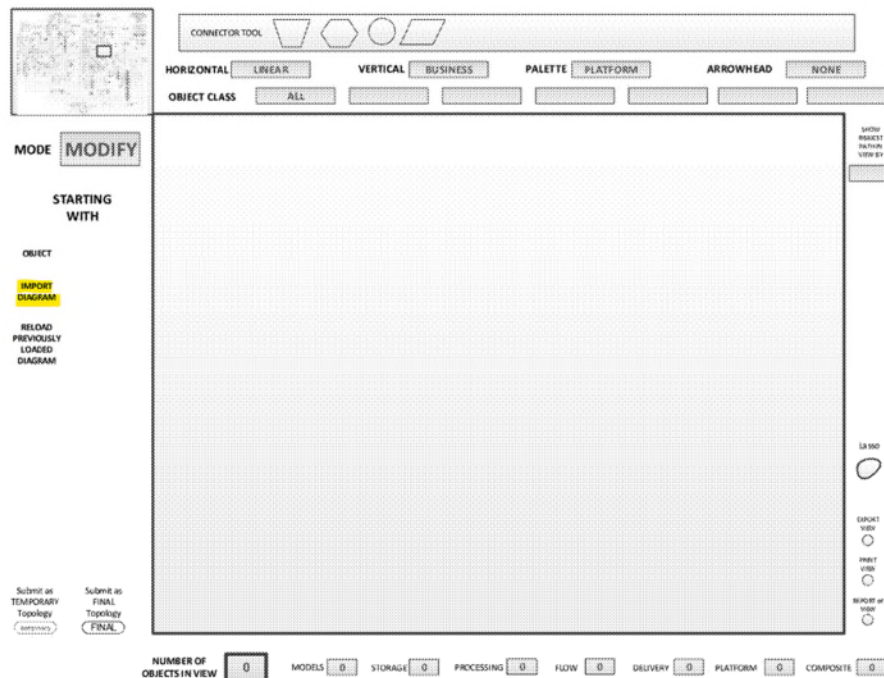


FIGURE 12

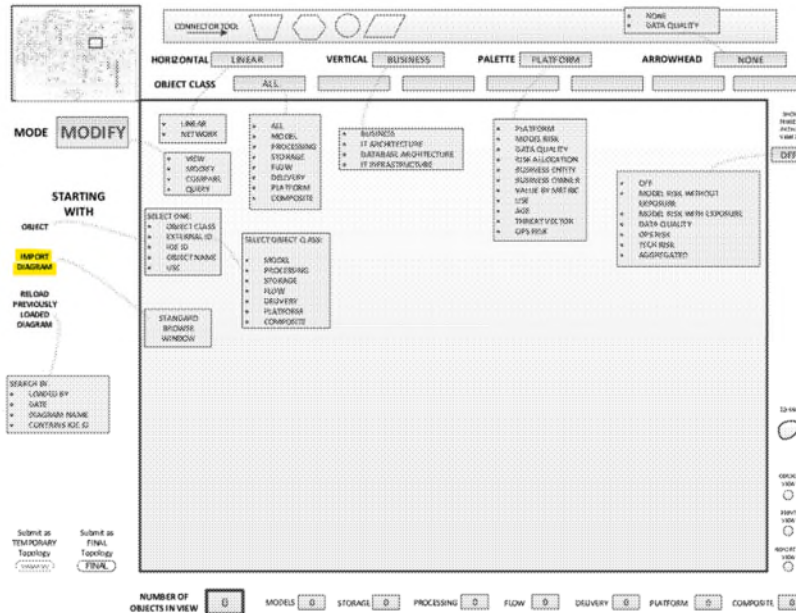


FIGURE 13

123. A POSITA would understand “the import function in the IOE GUI” to be an import interface displayed on a computing device as claimed because it allows a user to import third party diagram components, *e.g.*, Visio shapes and formats, which are included in a data file. A POSITA would further understand that Keenan’s import interface initiates reading of the data file because Keenan’s system checks for existing objects and forms new relationships after the user selects the import function.

- e. **Element 8[d]. “an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;”**

124. In my opinion, Keenan discloses this element. See Element 1[g].

Keenan discloses “an interface” because Keenan allows a “user to interact with the

IOE via a Graphical User Interface (GUI),” such as by selecting the import function. EX-1004 at 11:42-43; *see* Elements 1[e], 8[c].

125. Keenan further teaches that the interface is “configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files” because Keenan teaches a “map is preset that maps Visio shapes and formats to IOE shapes and formats. ... The existing Objects are then updated with new relationships based on the imported Visio diagram.” EX-1004, 18:36-50; *see also* Element 1[d]. The importation “initiate[s] a determination” of which threat model components are correlated with the third party diagram components in the mapping file because “existing objects are updated with new relationships” based on the importation of Visio diagrams—*i.e.*, the system must determine the relationships before updating new relationships.

- f. Element 8[e]. “a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

126. In my opinion, Keenan discloses this element. *See* Element 1[h]. A POSITA would understand that the Keenan’s disclosures discussed in Element 1[h] also disclose “a diagram interface” because Keenan discloses that “[d]iagrams

are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams ... serve[] as a communication medium to and from users and the IOE.” EX-1004, 2:3-6; *see also* display as “screen” *Id.* at 15:11-14; *see also* Element 1[h].

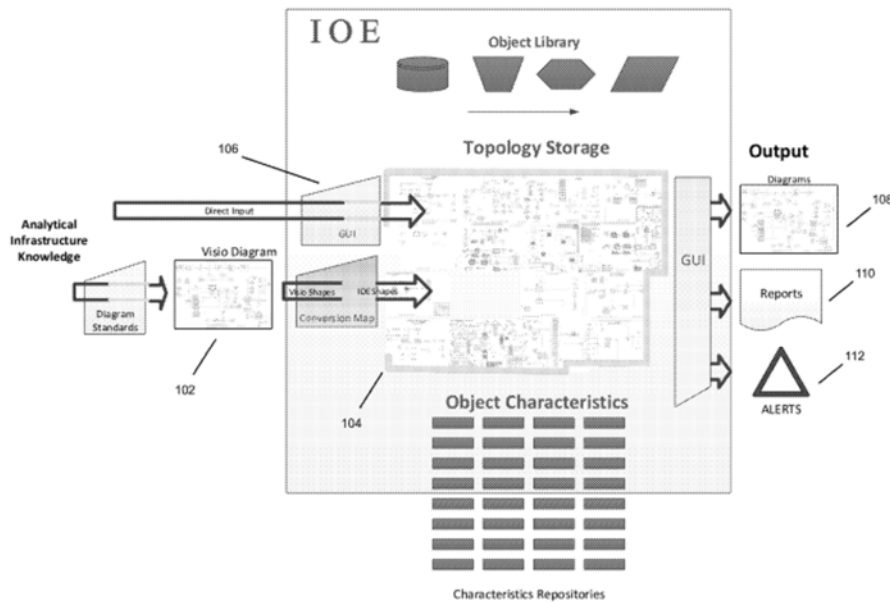


FIGURE 1

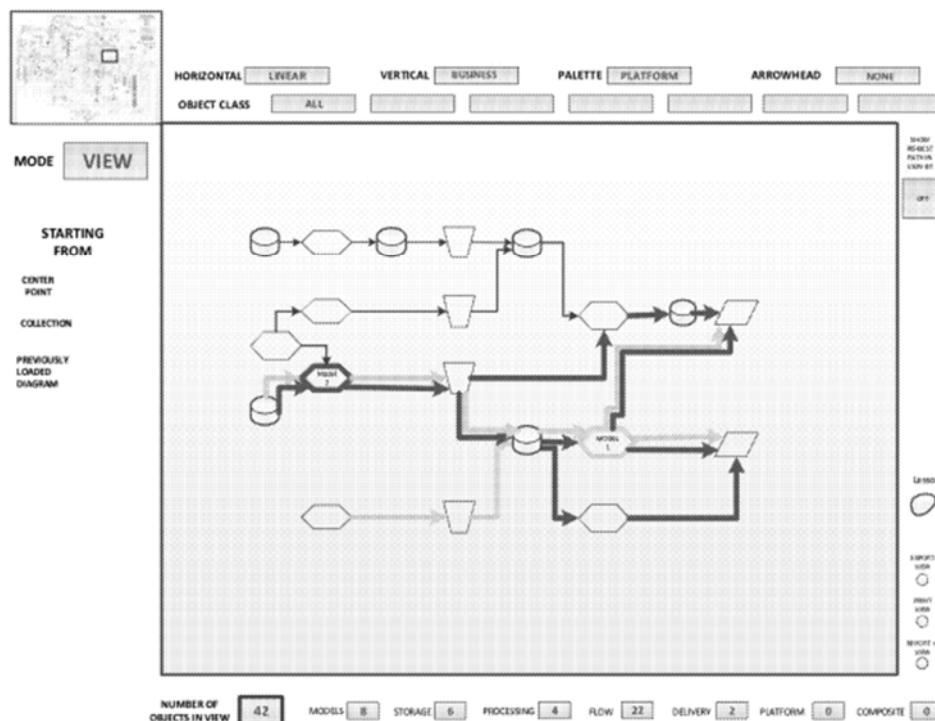


FIGURE 10

127. A POSITA would understand that this element is “displayed on the one or more computing devices.” See Element 1[e].

- g. **Element 8[f]. “a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

128. In my opinion, Keenan discloses this element. See Element 1[i]. A POSITA would understand that the Keenan’s disclosures discussed in Element 1[i] also disclose “a threat report interface” because Keenan discloses Views that are “exportable,”—i.e., interacted with through an interface, as reports.

129. Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” EX-1004, 5:40-44.

130. A POSITA would understand that this element is “displayed on the one or more computing devices.” See Element 1[e].

9. **Claim 9: “The system of claim 8, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”**

131. In my opinion, Keenan discloses this element. See Claim 2.

10. **Claim 10: The system of claim 8, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.**

132. In my opinion, Keenan discloses this element. See Claim 3.

11. **Claim 11: “The system of claim 8, wherein the data file comprises one of a VSD file, a VSDX file, a TMT File, an XML file, a JSON file, a PNG file, and a JPEG file.”**

133. In my opinion, Keenan discloses this element. See Claim 4.

12. **Claim 12: “The system of claim 8, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.”**

134. In my opinion, Keenan discloses this element. See also Claim 5.

Keenan teaches that “[e]ach object preferably has two identifiers: one internal to the IOE, and one that ties the object to an authoritative external repository. ...

Therefore, all ‘model’ objects in the IOE preferably carry their repository IDs as well as an ID internal to the IOE.” EX-1004, 6:12-26. A POSITA viewing

Keenan would understand that an “external ID” is a default title as claimed and a “name match” or “Name of object” is a user-defined name.

135. Keenan further discloses that “[t]o add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, ... Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found.” EX-1004, 18:51-60. A POSITA would understand that Keenan’s system allows for a user to add an object by correlating user-defined names of the third party diagram components with the threat model components—*i.e.*, through “Name of object,” or correlating default titles of the third party diagram components with the threat model components—*i.e.*, through “External Object ID.” A POSITA would further understand that this correlation is through “a file” because files are required to perform computer functions such as correlations.

13. Claim 13: “The system of claim 8, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.”

136. In my opinion, Keenan discloses this element. *See also* Claim 12. A POSITA would understand that Keenan’s system allows for a user to add an object

by correlating user-defined names of the third party diagram components with the threat model components—*i.e.*, through “Name of object,” or correlating default titles of the third party diagram components with the threat model components—*i.e.*, through “External Object ID.” A POSITA would further understand that this correlation is through “a file” because files are required to perform computer functions such as correlations.

- 14. Claim 14: The system of claim 8, wherein, for each of the subset of third party diagram components for which no correlated threat model component is found in the one or more mapping files, the diagram interface displays a placeholder component indicative of a match not being found.**

137. In my opinion, Keenan discloses this element. See Claim 6.

- 15. Claim 15: The system of claim 8, further comprising one or more mapping interfaces displayed on the one or more computing devices and configured to, in response to receiving user input, modify the one or more mapping files.**

138. In my opinion, Keenan discloses this element. Keenan discloses that “[t]o add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object.

Alternatively, the user can modify the object(s) found.” EX-1004, 18:51-60.

Specifically, Keenan provides for “a Modify mode, where the user can add, change and delete Objects or Characteristics,” in the Views of the IOE GUI—thus modifying the mapping file. *Id.*, 11:46-47; *see also id.*, Abstract, 4:1-2, 11:61-64, 18:29-35, 19:7-16, 20:61-21:9, Figs. 12-13. A POSITA would understand that the mapping file is modified when the view is modified. A POSITA would understand that a mapping interface is configured to modify as claimed because Keenan discloses a “screen” for user interaction, as well as a preset map that “maps Visio shapes and formats to IOE shapes and formats.” *Id.*, 18:36-50.

139. Keenan further discloses mapping interface is configured to modify “in response to user inputs” because Keenan explicitly specifies that a user may make modifications. *Id.*, 11:46-47 (disclosing “a Modify mode, where the user can add, change and delete Objects or Characteristics.”).

140. A POSITA would understand that this element is performed “using one or more computing devices.” *See* Element 1[e].

16. Claim 16:

- a. Element 16[p]. “A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:”**

141. In my opinion, Keenan discloses this element. *See* Elements 1[p]-1[a], 1[e], and 8[p].

- b. Element 16[a]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;”**

142. In my opinion, Keenan discloses this element. See Element 1[a] and 8[a].

- c. Element 16[b]. “one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);”**

143. In my opinion, Keenan discloses this element. See Elements 1[c]-1[d] and 8[b].

- d. Element 16[c]. “an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by the one or more computing devices, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

144. In my opinion, Keenan discloses this element. See Element 8[c].

- e. Element 16[d]. “an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;”**

145. In my opinion, Keenan discloses this element. See Element 1[g] and 8[d].

- f. **Element 16[e]. “a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of a computing network, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

146. In my opinion, Keenan discloses this element. See Element 1[h] and 8[e].

- g. **Element 16[f]. “a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

147. In my opinion, Keenan discloses this element. See Element 1[i] and 8[f].

- 17. **Claim 17: “The system of claim 16, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”**

148. In my opinion, Keenan discloses this element. See Claim 2.

- 18. **Claim 18: The system of claim 16, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.**

149. In my opinion, Keenan discloses this element. See Claim 3.

- 19. **Claim 19: The system of claim 16, wherein the one or more mapping files includes a file correlating user-defined names**

of the third party diagram components with the threat model components.

150. In my opinion, Keenan discloses this element. *See* Claim 12.

20. Claim 20: “The system of claim 16, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.”

151. In my opinion, Keenan discloses this element. *See* Claim 13.

C. Ground 2: Zheng In View of Baker and Jones Renders Obvious Claims 1-4, 7-11, and 15-18.

152. In my opinion, all of the limitations of claims 1-4, 7-11, and 15-18 are found in a combination of Zheng, Baker and Jones (collectively, the “Combination”).

1. Claim 1

a. Element 1[p]. “A threat modeling method, comprising:”

153. In my opinion, the Combination discloses this preamble. Zheng discloses “a system that can dynamically produce interactive and predictive threat models...” *E.g.*, EX-1005, 1:37-45; *see also id.*, Title (“intelligent threat modeling and visualization”); Abstract (“The threat modeling module is adapted to interact with the visualization module to graphically depict on the computer display the one or more security threats on the map in a threat modeling layer. The system further includes a security controls module configured to implement security controls for mitigating the one or more security threats identified by the threat modeling

module.”), 1:24-33, 2:1-18, 2:25-29, 2:36-49, 5:6-24 (“In some embodiments, the visualization module 116 generates and renders the threat modeling layer 204 in a computer display to visualize predicted security objectives, threats and/or vulnerabilities of the application architecture in the base layer 202. The visualization can be rendered in the same application landscape as (e.g., overlays) the base layer 202. Generally, data in the threat modeling layer 204 can be organized into threat agents, vectors, and targets. The threats can also be categorized into different types. Each threat can be visually associated with at least one of (i) a threat actor application, (ii) a threat target application and (iii) a displayable vector connecting the two to represent and visualize a particular attack methodology. For example, a vector connection can be a data flow path between one or more cities or between one or more buildings. A threat can provide one-to-one mapping between a threat actor and a threat target, a one-to-many mapping, a many-to-one mapping or a many-to-many mapping, depending on the nature of the threat.”), 5:25-35 (“These security risks can be manually and/or automatically identified by a threat modeling module 124 that is configured to interact with the visualization module 116 and supply data to the visualization module 116 via the input 108 for graphically depicting the identified risks. In one example, identification of a security threat can be performed manually by security personnel.”), 6:13-26.

b. Element 1[a]. “providing one or more databases, the one or more databases comprising:”

154. In my opinion, the Combination discloses this element. Zheng discloses that “at least one database table is used to store data for each layer and for each possible combination/interaction of the layers. As shown in FIG. 3, data for the base layer 202 can be stored in a base layer database table 502. Data for one or more additional layers can be stored in their corresponding database tables 504.” EX-1005, 8:14-24 (“FIG. 3 shows an exemplary layer creation flow diagram 500 for visualizing various layers of the multi-layered structure 200 of FIG. 2. Information used by the security risk analysis system 100 for visualizing and predicting security risks can be stored in one or more database tables. Depending on which layer(s) are selected by a user for visualization, data corresponding to the selected layers (in addition to the base layer 202) can be retrieved from their corresponding database tables and configured to interface with data from the base layer 202 to add interactive controls to the applications in the base layer 202.”). Figure 3 depicts:

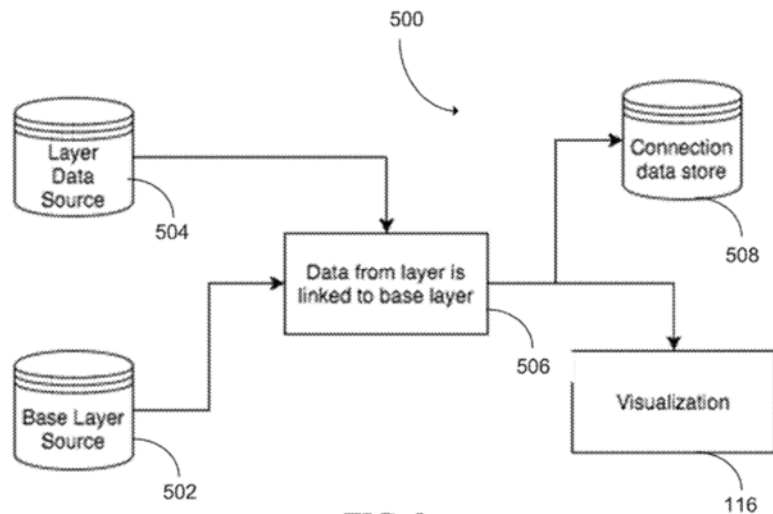


FIG. 3

See also *id.*, 8:25-9:10, 9:40-44, 10:35-39.

- c. **Element 1[b]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;”**

155. In my opinion, the Combination discloses this element. As discussed above, the PO defined “threat model components” as “relevant sources of threats.”

156. A “threat” in Zheng “visualizes each security threat as a threat actor, a threat target and a displayable vector connecting the threat actor to the threat target.” *Id.* 15:11-15. A “threat” in Zheng thus includes a “threat actor” and a “threat target,” and therefore, a POSITA would understand that a “threat” in Zheng includes both threats and threat model components as claimed by the ’366 patent.

157. Zheng discloses that “[e]ach threat can be visually associated with at least one of (i) a threat actor application, (ii) a threat target application and (iii) a

displayable vector connecting the two to represent and visualize a particular attack methodology.” EX-1005 at 5:14-18. Zheng’s “threat actor applications” are the claimed “plurality of threats,” and Zheng’s “threat target applications” are the claimed “plurality of threat model components” because Zheng discloses that “[t]he threat actor component 912 and the connections 914, 916, 918 and 920 can be a part of the threat modeling layer 204 of the multi-layered structure 200, where the connections 914, 916, 918 and 920 illustrate the impact of the threat actor 912 on various components of the base layer 202.” *Id.*, 12:1-3; *see also* 11:65-68. In other words, the threat target applications are relevant sources of threats (and therefore threat model components as claimed) impacted by the threats/threat actors (and therefore threats as claimed).

158. Zheng further discloses “wherein each of the threats is associated with at least one of the threat model components” because Zheng discloses that the threats (*i.e.*, Zheng’s threat actor applications) and threat model components (*i.e.*, Zheng’s threat target applications) are “visually associated” and that there is a “a displayable vector” that connects the threats and threat model components. *Id.*; *see also id.* 5:18-24 (“For example, a vector connection can be a data flow path between one or more cities or between one or more buildings. A threat can provide one-to-one mapping between a threat actor and a threat target, a one-to-many mapping, a many-to-one mapping or a many-to-many mapping, depending on the

nature of the threat.”); *Id.* at 2:36-42 (“In some embodiments, the threat modeling layer visualizes each security threat as a threat actor, a threat target and a displayable vector connecting the threat actor to the threat target. At least one of the threat actor or the threat agent is a component of the map. The displayable vector can visualize an attack methodology by the threat actor on the threat target.”).

159. These threat model components and threats are “stored” in and “associated ... through the one or more databases” because, as discussed in Element 1[a], Zheng discloses “at least one database table is used to store data for each layer and for each possible combination/interaction of the layers.” EX-1005, 8:25-27; *see also id.*, 8:25-9:10, 9:40-44, 10:35-39, Fig. 3.

d. Element 1[c]. “providing one or more mapping files communicatively coupled with the one or more databases,”

160. In my opinion, the Combination discloses this element. Zheng discloses that “[t]he present invention provides systems and methods for real-time (or near real-time) visualization of at least a portion of a company's application architecture and performing security risk analysis and threat detection based on the visualization.” Zheng accomplishes this visualization through “mapping techniques and concepts can be used in the context of data visualization to create a graphical representation of application connectivity and security.” EX-1005, 3:24-

34; *see also id.* 2:36-42 (“In some embodiments, the threat modeling layer visualizes each security threat as a threat actor, a threat target and a displayable vector connecting the threat actor to the threat target. At least one of the threat actor or the threat agent is a component of the map. The displayable vector can visualize an attack methodology by the threat actor on the threat target.”).

161. Zheng describes mapping between a threat actor and a threat target: “For example, a vector connection can be a data flow path between one or more cities or between one or more buildings. A threat can provide one-to-one mapping between a threat actor and a threat target, a one-to-many mapping, a many-to-one mapping or a many-to-many mapping, depending on the nature of the threat.” *Id.*, 5:18-24; *see also id.* 4:2-5 (“As described in detail below, the base layer 202 can visualize application components and relationships as a map to facilitate data extraction.”).

162. A POSITA would understand that Zheng’s “mapping techniques” and its “map” convey, and necessarily include, a “mapping file” as claimed because a mapping file would be required to perform mapping techniques.

163. A POSITA would further understand that the one or more mapping files in Zheng are communicatively coupled with the one or more databases because, as discussed in Element 1[a], Zheng discloses “at least one database table is used to store data for each layer and for each possible combination/interaction of

the layers.” EX-1005, 8:25-27; *see also id.*, 8:25-9:10, 9:40-44, 10:35-39, Fig. 3; Element 1[a].

- e. **Element 1[d]. “the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”); and”**

164. In my opinion, the Combination discloses this element. As discussed for Element 1[c], Zheng discloses a mapping file. And, Baker discloses that visual diagram components may be “of a third party software application (hereinafter ‘third party diagram components’)” as claimed and defined in the prosecution. Baker teaches that “risk management or control flowcharts may be imported at step 104 from third party software such as that marketed as Microsoft Office Visio®. The process 100 then proceeds to step 106. The text and shape of each element of the flowchart are stored in the process database 30 in memory 26. At step 106, the processor 20 acts to automatically assign or associate predetermined objectives or controls according to the shapes of the imported flowcharts. It is understood that the predetermined data may have been stored previously in the process database 30 of the memory 26. This process data may later be retrieved for use in initiating process searches over the internet 12 as will be described later. The process 100 then proceeds to step 108 at which assessment forms or other documentation may

be generated or stored for each process objective. Process is then received to step 52 of process 50 illustrated in FIG. 3.” EX-1006, ¶[0025].

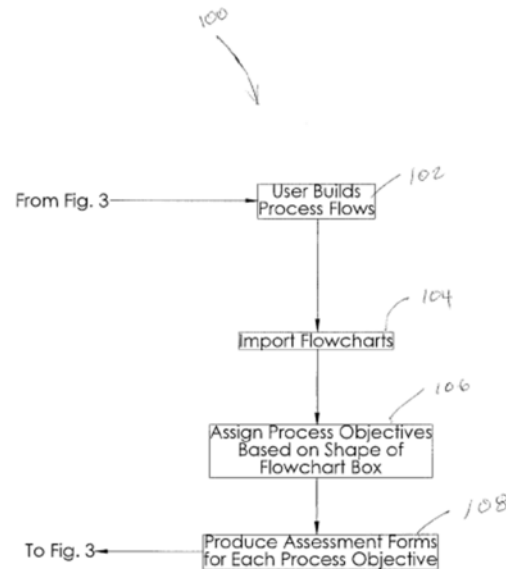


Fig. 4

Id., Fig. 4. A POSITA would understand that the Visio flowcharts are “of a third party software application” because Baker allows a user to select the option to “import” the Visio visual diagram components to Baker’s system, such that “a mapping file or the like as claimed in the claims,” such as the mapping file of Zheng, “would be useful to map between elements of the two software applications,” as defined in the file history. *E.g.*, EX-1004, ¶[0023].

165. Zheng even discloses importing from different “sources.” See EX-1005, at Fig. 1. This would motivate a POSITA to think about importing other items, such as Microsoft Visio diagrams as taught in Baker.

166. A POSITA would combine this teaching from Baker with the system of Zheng to enable the importation of diagrams from Microsoft Visio into the threat modeling software of Zheng such that the mapping file of Zheng would correlate the threat model components with visual diagram components from the Visio diagrams. Motivation for this combination is provided below.

- f. Element 1[e]. “in response to receiving one or more user inputs, using one or more user interfaces displayed on one or more computing devices communicatively coupled with the one or more databases:”**

167. In my opinion, the Combination discloses this element. Zheng discloses “in response to receiving more or more inputs, using one or more interfaces” because Zheng discloses that “[t]o provide for interaction with a user, the above described techniques can be implemented on a computing device in communication with a display device, e.g., a CRT (cathode ray tube), plasma, or LCD (liquid crystal display) monitor, a mobile computing device display or screen, a holographic device and/or projector, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse, a trackball, a touchpad, or a motion sensor, by which the user can provide input to the computer (e.g., interact with a user interface element). Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile

feedback; and input from the user can be received in any form, including acoustic, speech, and/or tactile input.” EX-1005, 13:9-18. Zheng also provides that “[f]or simulation processes, these threats, vulnerabilities, or attacks are available for a user to place, as one or more datasets 704, on an application, a group of applications, or a data flow link in the application ecosystem of the base layer 202 in the interactive visual interface.” EX-1005, 10:39-43; *see also id.* 13:30-37.

168. For the same reason, Zheng further discloses that the one or more interfaces is “displayed on one or more computing devices” because Zheng’s system “provide[s] for interaction with a user” by implementing the system on a “computing device in communication with a display,” such as a “user interface element.” EX-1005, 13:9-18. Zheng further discloses a system including “a visualization module for rendering, on a computer display, a map with components representative of the suite of software applications and relationships among the software applications.” *Id.*, 1:48-52. Moreover, Zheng discloses that its system may be implemented on, *e.g.*, a computer. *Id.*, 13:28-37.

169. A POSITA would understand that the “one or more computing devices” is “communicatively coupled with the one or more databases” because, as discussed in Element 1[a], Zheng discloses “at least one database table is used to store data for each layer and for each possible combination/interaction of the layers.” EX-1005, 8:25-27.

- g. **Element 1[f]. “using the one or more computing devices, reading a data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

170. In my opinion, the Combination discloses this element. As discussed above for Element 1[d], Baker discloses importing diagrams from Microsoft Visio into a threat modeling software application. Those diagrams are stored in a Visio file, which is a data file, that is generated by the Visio software application. Baker explains that the Visio file comprises a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components. See EX-1006, ¶[0025]; see also *id.*, ¶¶[0009], [0023], Fig. 4:

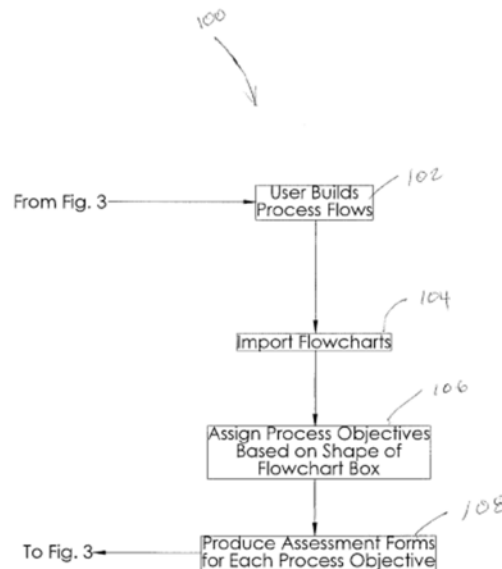


Fig. 4

171. A POSITA would understand that the Visio file (data file) defines one or more relationships between the subset of Visio (third party) components because the Visio components are imported as flowcharts, which are tools used to define relationships between components. A POSITA would further understand that the Visio file must be “read” using a computing device when imported because a file needs to be read to be imported.

172. A POSITA would further understand that Zheng discloses that the one or more interfaces “displayed on one or more computing devices” because Zheng’s system “provide[s] for interaction with a user” by implementing the system on a “computing device in communication with a display,” such as a “user interface element.” EX-1005, 13:9-18. Zheng further discloses a system including “a visualization module for rendering, on a computer display, a map with components representative of the suite of software applications and relationships among the software applications.” *Id.*, 1:48-52. Moreover, Zheng discloses that its system may be implemented on, *e.g.*, a computer. *Id.*, 13:28-37.

173. A POSITA would combine this teaching from Baker with the system of Zheng to enable the importation of diagrams from Microsoft Visio into the threat modeling software of Zheng such that the mapping file of Zheng would correlate the threat model components with visual diagram components from the Visio diagrams. Motivation for this combination is provided below.

h. Element 1[g]. “determining using the one or more computing devices, for the subset of third party diagram components, correlated threat model components as defined in the one or more mapping files;”

174. In my opinion, the Combination discloses this element. Zheng discloses “determining ... for the subset of third party diagram components, correlated threat model components,” because Zheng discloses a “map” that defines relationships between the third party components and among other system components. EX-1005, 2:38-42. Zheng further discloses that a threat target and a threat actor are associated together in a visualization display layers, where “new routes and/or applications” may be added to the base layer, and “and preexisting route(s) are present to connect the threat agent to the target application,” which may then “automatically predict new attack routes.” EX-1005, 5:35-37. In other words, Zheng’s system adds new suites of applications to the base layer and associates those applications to “target applications,” *i.e.*, correlated threat model components.

175. Zheng further discloses that this task is completed “using the one or more computing devices” because Zheng discloses that its system may be implemented on a computer. *See* Element 1[a].

176. Zheng further discloses that the correlated threat model components are “defined in the one or more mapping files” because Zheng discloses that “[a]

threat can provide one-to-one mapping between a threat actor and a threat target, a one-to-many mapping, a many-to-one mapping or a many-to-many mapping, depending on the nature of the threat.” EX-1005, 5:20-24. The “threat targets” or threat model components are therefore defined by the mapping file.

- i. **Element 1[h]. “displaying on the one or more user interfaces a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

177. In my opinion, the Combination discloses this element. Zheng discloses “displaying ... a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model” because Zheng discloses that “[a] threat can provide one-to-one mapping between a threat actor and a threat target, a one-to-many mapping, a many-to-one mapping or a many-to-many mapping, depending on the nature of the threat.” EX-1005, 5:20-24. Zheng’s mapping involves “visualizing in a computer display, as components in a base layer on a map, the suite of software applications and relationships among the software applications, automatically identifying one or more security threats in the suite of software applications, and graphically depicting on the computer display the one or more

security threats on the map in a threat modeling layer.” *Id.*, 2:3-10; *see also* 2:36-42 (“In some embodiments, the threat modeling layer visualizes each security threat as a threat actor, a threat target and a displayable vector connecting the threat actor to the threat target. At least one of the threat actor or the threat agent is a component of the map. The displayable vector can visualize an attack methodology by the threat actor on the threat target.”), Fig. 8.

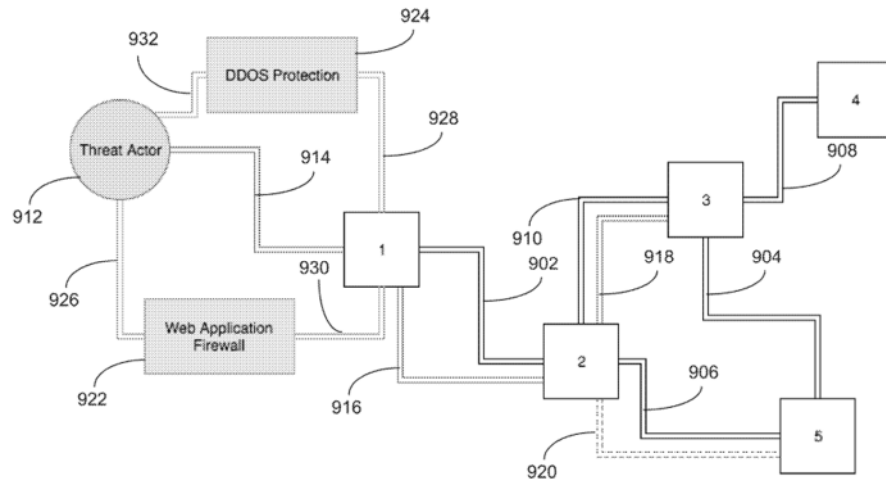


FIG. 8

178. Zheng further discloses that the displaying step is completed “on the one or more user interfaces” because of Zheng’s disclosures of “visualizing on a computer display” described above, and because Zheng discloses an “interactive visual interface.” See Element 1[e].

- j. **Element 1[i]. “generating, using the one or more computing devices, and displaying, on the one or more user interfaces, a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

179. In my opinion, the Combination discloses this element. For example,

Figure 1 of Jones discloses Reporter 114:

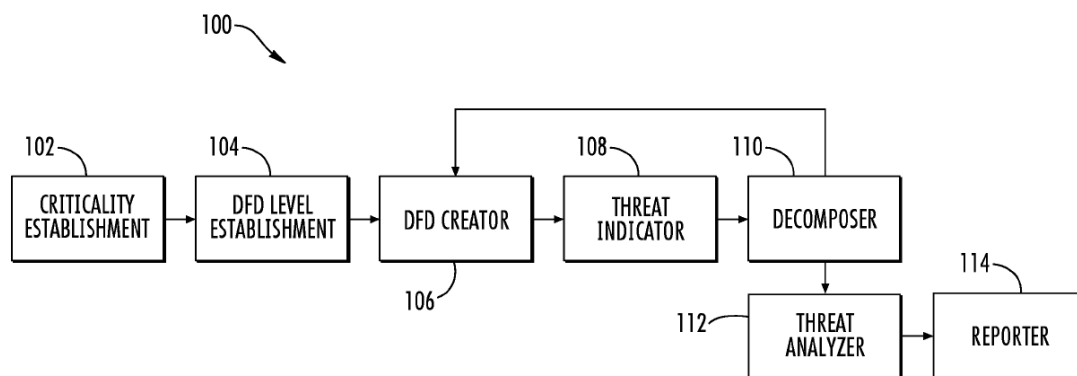


FIG. 1

180. Jones explains, “The reporter **114** may be configured to generate one or more reports from the threat analysis, at least one of which may include an organized arrangement of at least some of the attributes of at least some of the components, data flows or cybersecurity threat.” EX-1007, 6:59-63.

181. Figure 5 of Jones discloses Report Generator 502:

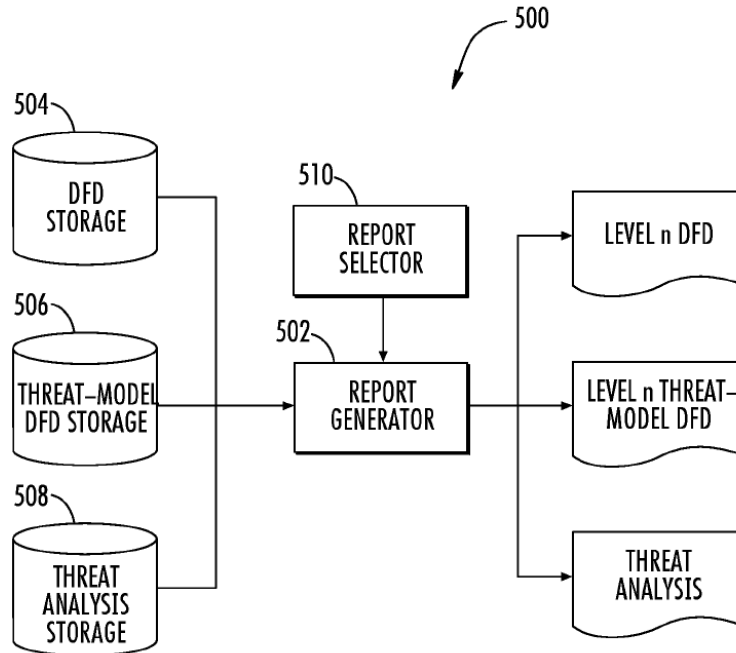


FIG. 5

182. Jones explains, “FIG. 5 illustrates a reporter **500**, which in some example implementations of the present disclosure may correspond to the reporter **114** of FIG. 1. . . . [T]he reporter may be configured to generate a report from the threat analysis, and including an organized arrangement of at least some of the attributes of at least some of the components, data flows or cybersecurity threat. . . . The report generator **502** may be configured to generate one or more reports based on the information retrieved from storage **504**, **506**, **508**. In some examples, the report generator may be coupled to a report selector **510** configured to select or receive selection of one or more reports to be generated by the report generator. These reports may include a graphical arrangement, textual arrangement or some combination of graphical and textual arrangement of the components, data

flows and/or cybersecurity threat for an information system.” EX-1007, 16:33-17:

67; *see also id.*, 2:54-63, 4:25-35, 6:59-63, 18:35-37, 19:29-45, Figs. 10A-10B.

Figs. 10A and 10B show example reports generated by Jones’ system:

1004

Numbering for this table – this does not correspond to numbering in the Visio diagram.	The weakness in the system or application that may be exploited. Can be at design, implementation, or operational level.	Event with potential adverse impact to the organization. Note that threats only need to be documented if it is deemed that there is a Current Risk.	Impact is independent of current controls or suggested mitigations.	Current security controls that mitigate the threat.	Probability that threat source will exploit the vulnerability given the current controls.	Risk given impact and current controls in place. This is an automatically calculated function of the impact and likelihood.
REF #	VULNERABILITY	THREAT	IMPACT	CURRENT CONTROLS	LIKELIHOOD	CURRENT RISK
1	Client-server connections unencrypted	Attacker may capture usernames and passwords.	Critical		Very High	Critical
2	RPC network connections not integrity-protected.	Attacker may change data using a man-in-the-middle attack.	Critical	Transmission occurs on a protected network segment available only to sys admins.	Medium	High
3	Buffer overflow finding in AppScan	Attacker may impersonate a system administrator.	Critical		Medium	High
4	Password authentication	System administrator may be impersonated.	High	Complex password policy	Medium	Medium
5	No failover system.	Failure of primary system will result in unavailability of functionality.	Medium		Medium	Medium
6	No vetting process for administrators	Improper access to system	High	Manager Approval	Low	Low
7	Admin actions not logged.	Rogue admin may subvert system with no forensic detection possible.	Low		Low	Low
8	Host-based firewall off	Increased attack surface of system.	Medium	Network-based firewall protects system.	Low	Low

FIG. 10A ● YELLOW ● PURPLE ● RED ● GREEN

1004

Numbering for this table – this does not correspond to numbering in the Visio diagram.	Security controls that are recommended to bring the Current Risk rating lower.	Risk given that suggested mitigations are implemented.	The data flow diagram (DFD) element(s) that the threat applies to.	Additional clarifying information. If a finding is preliminary, meaning it needs further research indicate “Preliminary” here.	Use to note whether the review process has resulted in acceptance of this threat and associated impact and risk ratings.
REF #	SUGGESTED MITIGATIONS	RESIDUAL RISK	APPLIES TO	REMARKS	REVIEW STATUS
1	Encrypt client-server connections	Medium	Data flows between client app and server.	This is an information disclosure threat that may lead to a follow-on threat of spoofing threat.	Reviewed
2	Protect RPC connections with a protocol that provides integrity-protection (like SSL/TLS)	Medium	Master-to-Backup data flow		Likelihood rating not agreed
3	Fix buffer overflow.	Medium	Login process	CYSS rating of threat indicated high difficulty to reproduce, which is why likelihood is rated Medium	Not Reviewed
4	Use strong two-factor authentication.	Low	Database system	Preliminary: <Describe why finding is preliminary>	Waiting to clear Preliminary status
5	Implement a failover server in the same datacenter as primary server. Implement regional failover server.	Medium	Web server Application server	Preliminary: <Describe why finding is preliminary> Regional failover needed to bring residual risk to Low.	Waiting to clear Preliminary status
6	Background check	Low	Top level sys admins	Not a traditional STRIDE threat. Improper admin designation could result in all STRIDE threats.	Reviewed
7	Log admin success and fail actions.	Low	Application Server		Reviewed
8	Turn on host-based FW	Low	Web server	Type is “Undetermined” as the vulnerability is very general.	Not Reviewed

● YELLOW ● GREEN FIG. 10B

183. A POSITA would combine this teaching from Jones with the system of Zheng to enable the user to view a report of the threat model. Motivation for this combination is provided below.

2. Claim 2: “The method of claim 1, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”

184. In my opinion, the Combination discloses. Zheng discloses that its system may be entirely implemented by a computer. See Element 1[a]. “The above-described techniques can be implemented in a distributed computing system that includes a back-end component. The back-end component can, for example, be a data server, a middleware component, and/or an application server. The above described techniques can be implemented in a distributed computing system that includes a front-end component. The front-end component can, for example, be a client computer having a graphical user interface, a Web browser through which a user can interact with an example implementation, and/or other graphical user interfaces for a transmitting device. The above described techniques can be implemented in a distributed computing system that includes any combination of such back-end, middleware, or front-end components.” EX-1005, 13:28-37.

3. Claim 3: The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.

185. In my opinion, the Combination discloses this element. Zheng teaches that “FIG. 3 shows an exemplary layer creation flow diagram 500 for visualizing various layers of the multi-layered structure 200 of FIG. 2. Information used by the security risk analysis system 100 for visualizing and predicting security risks can be stored in one or more database tables.” EX-1005, 8:14-18; see also Element 1[a] (noting Zheng’s disclosure that “at least one database table is used to store data for each layer and for each possible combination/interaction of the layers.”).

4. Claim 4: “The method of claim 1, wherein the data file comprises one of a VSD file, a VSDX file, a TMT file, an XML file, a JSON file, a PNG file, and a JPEG file.”.

186. In my opinion, the Combination discloses this element. See Element 1[e]. A POSITA would understand that a Visio file is a “VSD file” or a “VSDX file” and that Baker therefore discloses importing a VSD or VSDX file. For example, https://en.wikipedia.org/wiki/Microsoft_Visio states that Visio uses the VSD and VSDX file formats, which is consistent with my own knowledge and experience.

5. **Claim 7: “The method of claim 1, further comprising determining, for the one or more relationships defined by the data file, correlated threat model components as defined in the one or more mapping files.”**

187. In my opinion, the Combination discloses this element. *See* Elements 1[g]-1[f]. Zheng discloses, referring to Fig. 8, that “the numbered boxes represent applications within the ecosystem and the connections 902, 904, 906, 908, 910 represent relationships/data flows among the applications. The threat actor component 912 and the connections 914, 916, 918 and 920 can be a part of the threat modeling layer 204 of the multi-layered structure 200, where the connections 914, 916, 918 and 920 illustrate the impact of the threat actor 912 on various components of the base layer 202.” EX.-1004, 11:59-68;12:1-3.

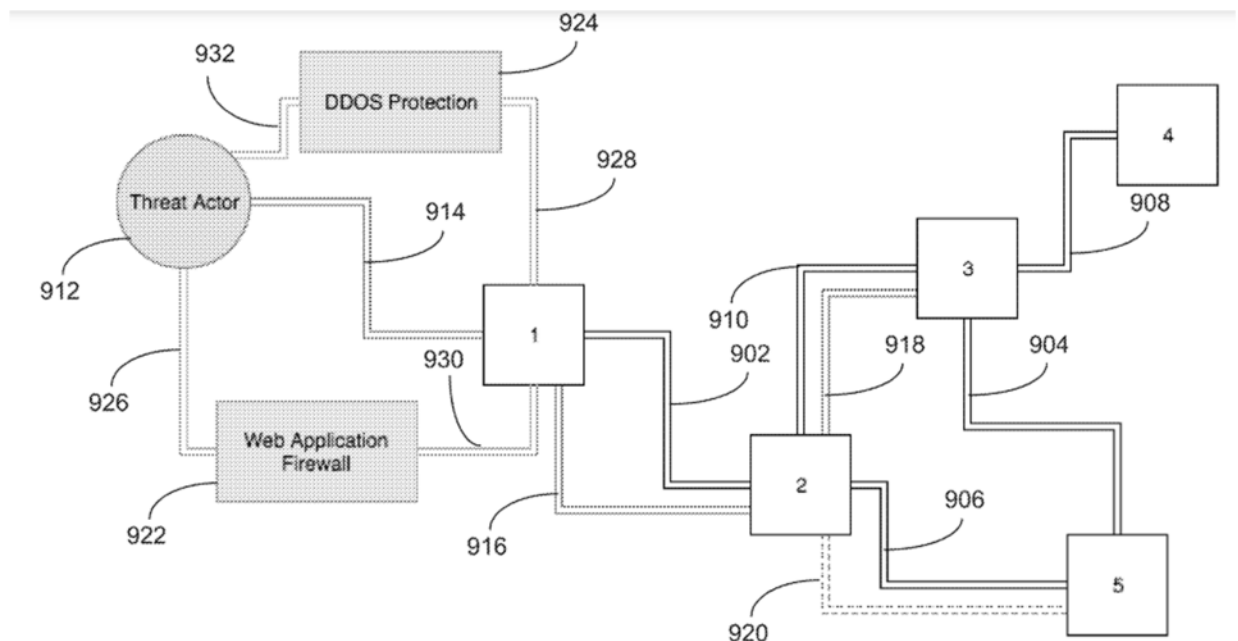


FIG. 8

6. Claim 8:

- a. **Element 8[p]. “A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:”**

188. In my opinion, the Combination discloses this element. A POSITA would understand that Zheng’s one or more databases is communicatively coupled with one or more computing devices because Zheng discloses that its methods and systems are implemented by a computer and that “at least one database table is used to store data for each layer and for each possible combination/interaction of the layers.” *Id.*, 21:26-33; Elements 1[p]-1[a] and 1[e]

- b. **Element 8[a]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;”**

189. In my opinion, the Combination discloses this element. *See supra*, Elements 1[b].

- c. **Element 8[b]. “one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);”**

190. In my opinion, the Combination discloses this element. *See supra*, Elements 1[c]-1[d].

- d. **Element 8[c]. “an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by one of the one or more computing devices, the data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

191. In my opinion, the Combination discloses this element. *See supra*, elements 1[e] and 1[f]. Baker discloses an import interface displayed on the screen of the computing device to initiate the reading of the Visio file in response to a user input. Baker teaches that “risk management or control flowcharts may be imported at step 104 from third party software such as that marketed as Microsoft Office Visio®. The process 100 then proceeds to step 106. The text and shape of each element of the flowchart are stored in the process database 30 in memory 26. At step 106, the processor 20 acts to automatically assign or associate predetermined objectives or controls according to the shapes of the imported flowcharts. It is understood that the predetermined data may have been stored previously in the process database 30 of the memory 26. This process data may later be retrieved for use in initiating process searches over the internet 12 as will be described later. The process 100 then proceeds to step 108 at which assessment forms or other documentation may be generated or stored for each process

objective. Process is then received to step 52 of process 50 illustrated in FIG. 3.”

EX-1006, ¶[0025].

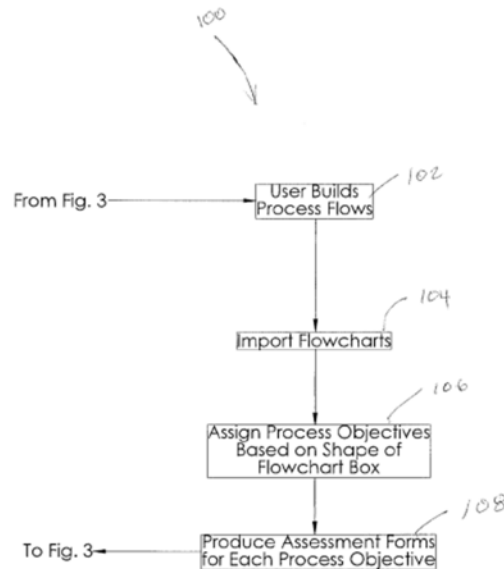


Fig. 4

Id., Fig. 4.

192. A POSITA would understand import function to be an import interface displayed on a computing device as claimed because it allows a user to import third party diagram components, *e.g.*, Visio flowcharts, which are included in a data file. A POSITA would further understand that Baker's import interface initiates reading of the data file when imported.

- e. **Element 8[d]. “an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;”**

193. In my opinion, the Combination discloses this element. See Elements 1[g], 1[d], 1[e]. Zheng discloses “an interface” because Zheng discloses an “interactive visual interface.” EX-1005, 10:39-43; Element 1[e].

194. Zheng further teaches that the interface is “configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files” because Zheng discloses a “map” that defines relationships between the third party components and among other system components. EX-1005, 2:38-42. Zheng further discloses that a threat target and a threat actor are associated together in a visualization display layers, where “new routes and/or applications” may be added to the base layer, and “and preexisting route(s) are present to connect the threat agent to the target application,” which may then “automatically predict new attack routes.” EX-1005, 5:35-37. In other words, Zheng’s system adds new suites of applications to the base layer and associates those applications to “target applications,” *i.e.*, correlated threat model components. Element 1[g].

- f. **Element 8[e]. “a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

195. In my opinion, the Combination discloses this element. *See supra* Element 1[h]. A POSITA would understand that the Zheng's disclosures discussed in Element 1[h] also disclose "a diagram interface" because Zheng discloses "graphically depicting on the computer display the one or more security threats on the map in a threat modeling layer," and allowing the user to interact with the display. EX-1005, 2:3-10, 2:36-42, Fig. 8, 8:14-24; Element 1[h].

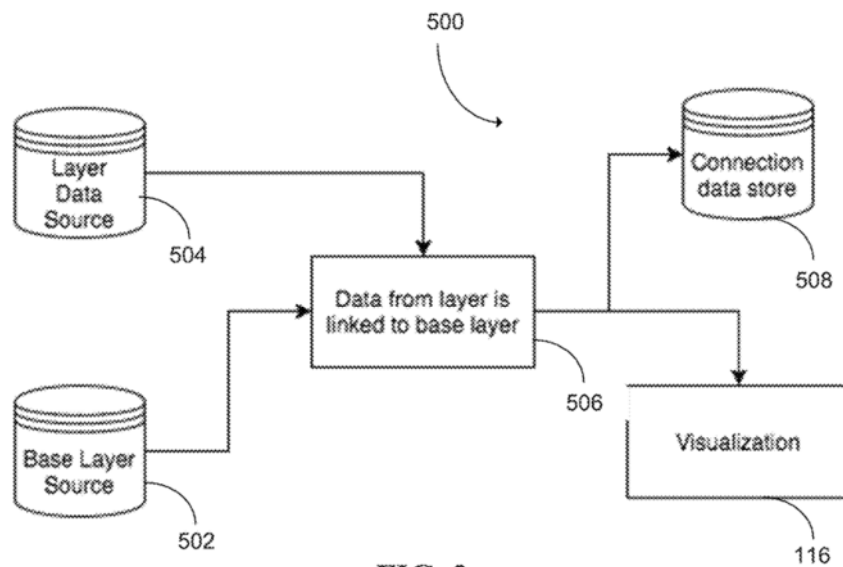


FIG. 3

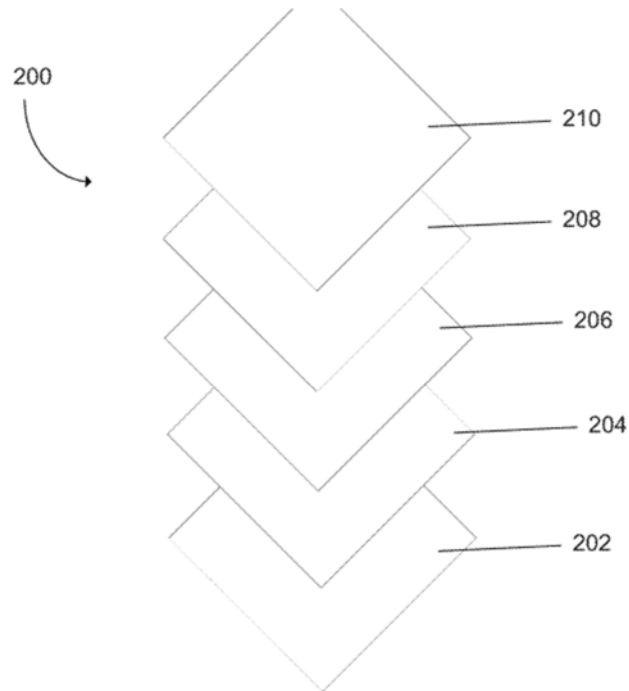


FIG. 2

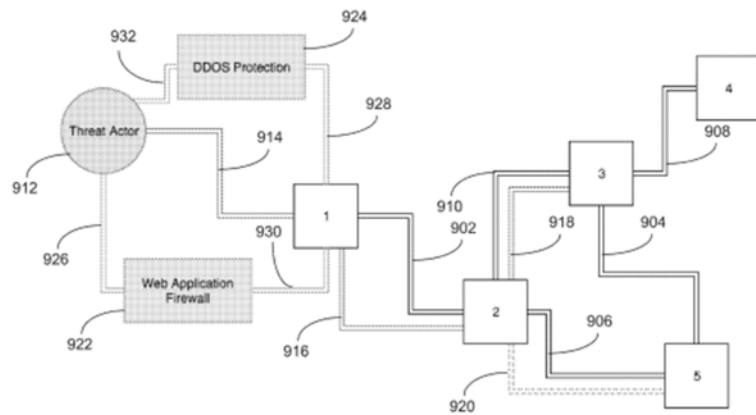


FIG. 8

196. A POSITA would understand that this element is “displayed on the one or more computing devices.” See Element 1[e].

- g. Element 8[f]. “a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

197. In my opinion, the Combination discloses this element. *See supra* Element 1[i]. A POSITA would understand that the Jones’ disclosures discussed in Element 1[i] also disclose “a threat report interface” because Jones discloses that “FIG. 5 illustrates a reporter 500, which in some example implementations of the present disclosure may correspond to the reporter 114 of FIG. 1. ... [T]he reporter may be configured to generate a report from the threat analysis, and including an organized arrangement of at least some of the attributes of at least some of the components, data flows or cybersecurity threat...” EX-1007, 16:33-17:41. A POSITA would understand that Jones’ reporter configured to generate a report by a user is a threat report interface as claimed.

- 7. Claim 9: “The system of claim 8, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”**

198. In my opinion, the Combination discloses this element. *See* Claim 2.

- 8. Claim 11: The system of claim 8, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.**

199. In my opinion, the Combination discloses this element. *See* Claim 4.

- 9. Claim 15: The system of claim 8, further comprising one or more mapping interfaces displayed on the one or more**

computing devices and configured to, in response to receiving user input, modify the one or more mapping files.

200. In my opinion, the Combination discloses this element. Zheng discloses “one or more mapping interfaces displayed on the one or more computing devices.” See Element 1[e], 1[h]. Zheng further discloses “one or more mapping interfaces ... configured to, in response to receiving user input, modify the one or more mapping files” because Zheng discloses that a “user can interact with the resulting visualization” of the user interface to modify the display, *i.e.*, “to achieve the desired display effects.” For example, Zheng discloses that “the threat modeling layer 204, the security controls layer 206, or the threat update layer 208 has sub-categories in the layer that can be toggled (e.g., enabled or disabled) by the user,” *e.g.*, “a particular type of threats in the threat modeling layer 204 can be toggled on or off to provide more focused visualization.” EX-1005, 11:46-49; *see also id.* 11:49-55 (“In some embodiments, the visualization module 116 allows panning and/or zooming of the visualized structure. Depending on what information is graphically illustrated in the computerized display window, the visualization module 116 can display different information and/or hierarchies of information upon receiving a panning or zooming command from a user.”). A POSITA would understand that modifications to the display likewise modify the mapping file.

10. Claim 16:

- a. Element 16[p]. “A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:”**

201. In my opinion, the Combination discloses this element. See Elements 1[p]-1[a].

- b. Element 16[a]. “a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;”**

202. In my opinion, the Combination discloses this element. See Element 1[a].

- c. Element 16[b]. “one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);”**

203. In my opinion, the Combination discloses this element. See Elements 1[c]-1[d].

- d. Element 16[c]. “an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by the one or more computing devices, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;”**

204. In my opinion, the Combination discloses this element. *See* Element 8[c].

- e. **Element 16[d]. “an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;”**

205. In my opinion, the Combination discloses this element. *See* Element 1[g].

- f. **Element 16[e]. “a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of a computing network, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and”**

206. In my opinion, the Combination discloses this element. *See* Element 1[h].

- g. **Element 16[f]. “a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.”**

207. In my opinion, the Combination discloses this element. *See* Element 1[i].

11. Claim 17: “The system of claim 16, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.”

208. In my opinion, the Combination discloses this element. *See* Claim 2.

12. Claim 18: The system of claim 16, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.

209. In my opinion, the Combination discloses this element. *See* Claim 3.

13. A POSITA Would Have Been Motivated to Combine Zheng with Baker and Jones.

210. A POSITA would have been motivated to combine Zheng, Baker, and Jones. First, like the '366 patent, all three references relate to the same general subject matter of generating threat models on a computer. *See* EX-1005, Abstract; EX-1006, Abstract, EX-1007, Abstract.

211. Second, Zheng provides express motivation when it discusses the need to minimize the tedious burden on human beings that exists in certain prior art threat modeling software applications:

With the growing complexity of application architecture and application infrastructure, it becomes tedious and often unmanageable to analyze application security across a large portion or an architectural stack. Today's threat modeling applications are often static and require either manual input or antiquated approaches to account for new architecture, code and/or infrastructure. Today's threat modeling approaches are also ineffective and unable to scale to hundreds or thousands of software applications in a complex, connected software application ecosystem. . . . ***Accordingly, there is a need for improved systems, methods and apparatuses for a***

***more effective and efficient threat modeling approach
in the context of application architecture and
infrastructure.***

EX-1005, 1:21-34.

212. Adding the teachings of Baker and Jones to the system disclosed in Zheng would further Zheng's stated goal of an improved threat modeling system. For example, Zheng recognizes the need for improved threat modeling systems "that can dynamically produce interactive and predictive threat models that can support real-time (or near real-time) risk analytics based on automated data aggregation and transformation, easy-to-understand visualization." EX1005, 1:37-45. In other words, Zheng recognizes the need to produce threat models faster (in real time or near real time) and using easy-to-understand visualizations. Thus, a POSITA would be motivated to combine these references because the ability to import existing diagrams from Visio, as taught in Baker, and the ability to generate threat reports, as taught in Baker and Jones, would reduce amount of tedious and unmanageable work that human operators would otherwise need to handle and provide commonplace, easy-to-understand visualizations. *See also* EX-1011, p. 94.

213. Third, Microsoft Office Visio was a well-known software tool for generating flowcharts and diagrams at the time the patents were filed. *E.g.*, EX-1007, 18:30-43 ("In some examples, the GUI may be provided by a commercially-

available diagramming tool such as Microsoft Visio®, specifically configured to carry out various operations of example implementations; and in some examples, the report may be output to and displayed by a commercially-available spreadsheet tool such as Microsoft Excel®.”). A POSITA would have been motivated to allow users to take existing diagrams already prepared in Visio and import those diagrams into the system of Zheng. Users were already comfortable with Visio, which has many good features and is easy to use. This would allow users to create threat models from diagrams already created in Visio to save time of creating those same diagrams using a native interface in the Zheng system. If a user prefers Visio he or she could simply continue using that software rather than learning how to use the native interface of Zheng.

214. In general, compatibility between software programs is a positive improvement that users appreciate. *E.g.*, EX-1007, 18:30-43 (noting compatibility between Jones’ system and Microsoft Visio). For example, it is commonplace today to use one program, such as Adobe Acrobat, to create a pdf and to import that pdf into a second program such as a Microsoft PowerPoint presentation. *Id.* (discussing Jones’ system is also compatible with Excel and other commonplace programs). Similarly, users would benefit from being able to import a Visio diagram into the threat modeling tool of Zheng. For example, a POSITA would

readily understand that Zheng's mapping file would be useful in mapping the relationships between the imported Visio diagrams.

215. Fourth, the threat model reports of Jones are a useful feature. Reports are efficient mechanisms for humans to digest complicated sets of data. A POSITA would have been motivated to combine Zheng with the report generation feature of Jones so that a user of the Zheng system could have reports to quickly understand threat models and potential threats to underlying systems.

216. Combining these references would have been within the skillset of a POSITA at the time of the invention and would require no undue experimentation. Further, such a combination would have an easily predictable result. Accordingly, a POSITA would have had a reasonable expectation of success.

D. Ground 3: The References of Ground 2 in Further View of Galliano Render Obvious Claims 2, 9, and 17

217. In my opinion, claims 2, 9, and 17 are rendered obvious by combining the references of Ground 2 with Galliano. Specifically, Galliano discloses the limitations of claims 2, 9, and 17 in Figure 3, which states that the “[m]odel is either stored in a back-end repository (ex. database) or in memory within a computing system”:

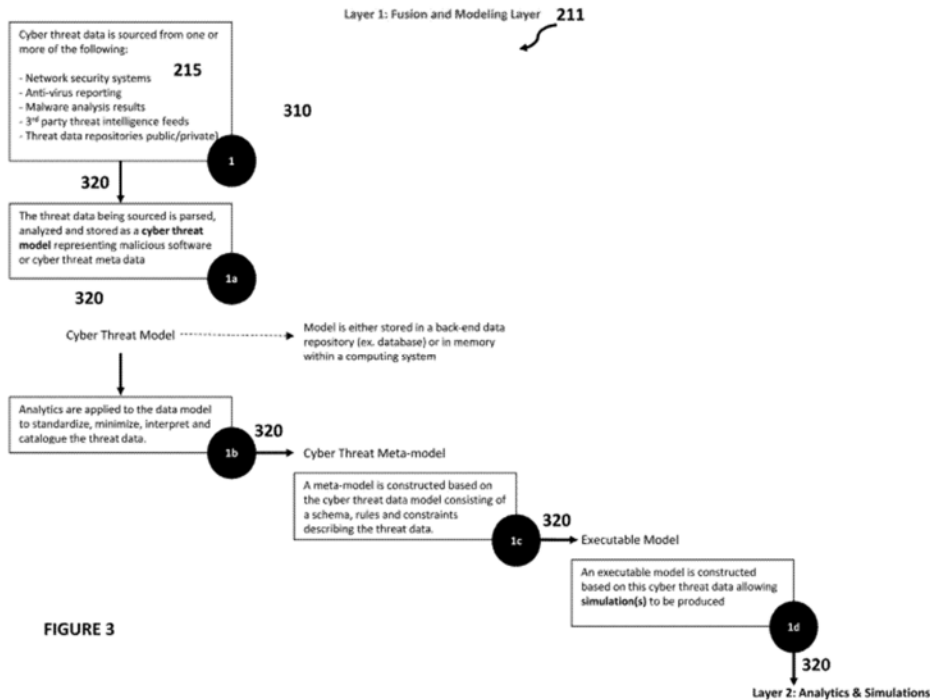


FIGURE 3

EX-1008, Figure 3.

218. A POSITA would have been motivated to combine Zheng, Baker, Jones, and Galliano for the same reasons discussed in Ground 2 for Zheng, Baker, and Jones. Galliano also relates to the general subject matter of generating threat models on a computer. See Ground 2; EX-1008 at Abstract.

219. Furthermore, a POSITA would have been specifically motivated to combine the teachings of Galliano with the system of Zhang. Galliano teaches that mapping files can be stored in a database or in a computing device without using a database. EX-1008, Figure 3. This would have been immediately clear to a POSITA at the relevant time, and a POSITA would understand that mapping files need to be stored and that as a matter of mere design choice, they could be stored

in a database or in a mechanism that does not require a database, such as a file system. A POSITA might choose to use a database in some situations and a file system in others. For example, databases are efficient tools when there the volume of data is significant and if search capabilities are desired, as might be the case if there are thousands of mapping files. However, databases also require database software, which would add to the overall cost of the underlying a system. A POSITA therefore might choose to store the mapping files in a file system and not in a database to save on the costs of the overall system or if there are not many mapping files being used or if search capability is not important.

220. Combining the references of Ground 2 with Galliano would have been within the skillset of a POSITA at the time of the invention and would require no undue experimentation. Further, such a combination would have an easily predictable result. Accordingly, a POSITA would have had a reasonable expectation of success.

E. Ground 4: The References of Ground 2 in Further View of Keenan Render Obvious Claims 1-20

221. In my opinion, claims 5, 6, 12, 13, 14, 19, and 20 are rendered obvious by the references of Ground 2 in further view of Keenan. Claims 5, 6, 12, 13, 14, 19, and 20 describe the user of certain names and titles for third-party components in the threat model. As shown above and in Exhibit B, Keenan

expressly discloses the limitations of those claims, as well as all other Challenged Claims.

222. A POSITA would have been motivated to combine the teachings of Keenan with the system of Zheng for the reasons discussed above for Ground 2. In addition, Keenan describes a certain naming convention for names and titles for third-party components. These are mere design choices that would have been obvious to a POSITA, and they are expressly disclosed in Keenan. A POSITA would understand that names and titles were required, and Keenan discloses this specific aspect.

223. Combining Zheng with Baker, Jones and Keenan would have been within the skillset of a POSITA at the time of the invention and would require no undue experimentation. Further, such a combination would have an easily predictable result. Accordingly, a POSITA would have had a reasonable expectation of success.

F. Ground 5: The References of Ground 2 in Further View of the Knowledge of a POSITA Render Obvious Claims 2, 5, 6, 9, 12-14, 17, 19 and 20

224. In my opinion, claims 2, 5, 6, 9, 12-14, 17, 19, and 20 also are rendered obvious by the references of Ground 2 (Zheng, Baker, and Jones) in further view of the knowledge of a POSITA.

225. A POSITA would have understood that mapping files can be stored in a database or in a computing device without using a database as specified in claims 2, 9, and 17. This would have been immediately clear to a POSITA at the relevant time, and a POSITA would understand that mapping files need to be stored and that as a matter of mere design choice, they could be stored in a database or in a mechanism that does not require a database, such as a file system. A POSITA might choose to use a database in some situations and a file system in others. For example, databases are efficient tools when there the volume of data is significant and if search capabilities are desired, as might be the case if there are thousands of mapping files. However, databases also require database software, which would add to the overall cost of the underlying a system. A POSITA therefore might choose to store the mapping files in a file system and not in a database to save on the costs of the overall system or if there are not many mapping files being used or if search capability is not important.

226. A POSITA would have understood that third-party components can be named or titled in a certain way as specified in claims 5, 6, 12-14, and 19-20 as a matter of mere design choice.

227. Modifying Zheng, Baker, and Jones in this manner with the knowledge of a POSITA would have been within the skillset of a POSITA at the time of the invention and would require no undue experimentation. Further, such a

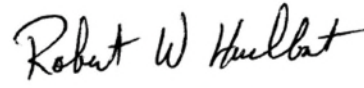
combination would have an easily predictable result. Accordingly, a POSITA would have had a reasonable expectation of success.

XI. CONCLUSION

228. Claims 1-20 of the '366 patent are invalid for the reasons discussed above and in the attached Exhibit B.

229. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both under Section 1001 of Title 18 of the United States Code.

Executed on March 7th, 2023 at 8:50 AM, Enfield, CT, USA.

A handwritten signature in black ink that reads "Robert W. Hurlbut". The signature is written in a cursive style with a horizontal line extending from the end of the name.

Robert Hurlbut

EXHIBIT A

ROBERT W. HURLBUT
Principal Application Security Architect / Threat Modeling Lead
Enfield, CT

Phone: 860-515-8459, E-mail: Robert at RobertHurlbut.com

SUMMARY

High performance Principal Application Security Architect and Threat Modeling Lead with outstanding experience as an Application Security Architect / Threat Modeling Architect and Lead / Technical Lead / Developer Trainer & Mentor in diverse arenas, from Fortune 50 organizations to start-ups. Keen eye for detail, with the ability to *concentrate on the big picture while* managing day-to-day operations and *maintaining focus on the company's mission, goals, and revenue*. *Proactive and aggressive in adding value through continuously evaluating and recommending process improvements that streamline operations and processes and lead to significant cost / time reductions.*

- Proven ability in developing concepts and strategies; defining scope, procedures and objectives; managing resources, and interpreting and documenting results.
- Delivering customized training for developers, architects, testers, project managers, and other stakeholders in software / application security topics.
- In-depth understanding of multiple development languages and environments.
- Outstanding ability to drive innovative solutions to complex problems, delivering continuous improvements in performance, cost, quality, and security.
- Strong leader with superior communication skills and the ability to articulate requirements, respond to questions, and convey complex technical information in clear and understandable terms to any audience.
- Calm, flexible, and focused in urgently-paced, demanding, and time-sensitive environments.

PROFESSIONAL COMPETENCIES

- | | | |
|---|-----------------------|------------------------|
| • Security Infrastructures and Patterns | ♦ Secure Coding | ♦ Government |
| • Web, Mobil, and Cloud Security | ♦ Penetration Testing | ♦ Training / Education |
| • Mobile Architecture / Development | ♦ Threat Modeling | ♦ Security Auditing |
| • Object-Oriented Analysis / Design | ♦ DevOps / Agile | ♦ Microservices |
| • Database Architecture and Development | ♦ Project Management | ♦ Problem Resolution |
| • Security / Application Architecture / Development | ♦ DDD / MVC / MVVM | ♦ Microsoft MVP |

PLATFORMS, LANGUAGES & APPLICATIONS

Platforms Windows, Windows Server, Citrix (Terminal Server / Remote Desktop), Linux / UNIX, Mac
Languages C++, C#/.NET, VB, Java, UML, SQL, Python
Databases Microsoft SQL Server, Oracle
Modeling Microsoft Visio, MS Threat Modeling Tool, Threat Modeler, Irius Risk
Security HP Fortify, Checkmarx, Threadfix, SD Elements, Burp Suite, ZAP Proxy, Kali Linux, Gauntlt
Reporting & Project Management Reporting Services for SQL Server, VSView, Microsoft Project

EXPERIENCE

11/21 – Present Aquia, Inc., Enfield, CT

Principal Application Security Architect / Threat Modeling Lead

- Mural, Miro, Lucid Chart, Confluence, Jira
- Leading Threat Modeling Program at Aquia and for various clients including a large Federal Government Agency
- Providing training to many developer teams

9/2017 – 11/2021 Bank of America, Enfield, CT

Senior Vice President, Threat Modeling Senior Architect / Lead

- Microsoft Threat Modeling Tool 2020, Microsoft Visual Studio, Python, C#/.NET.
- Led and further developed Threat Modeling Program at Bank of America which was started in 2016 (see below) as part of GIS CST Architecture (2017- 2021) and then GIS BISO CSW (2021).

- Provided training to 6,000+ associates in In-Person and Virtual sessions. Co-wrote Threat Modeling training that is required in 2021 by 25,000+ developers in the bank.
- Started and led the Threat Modeler in Residence Program from 2019 to 2021. This was a volunteer program of 25+ associates who helped project teams build threat models (essentially a Security Champions Program).
- Worked with project teams to build over 1,000+ threat models over three years as well as empowered others on the GIS CST Architecture Team and other security personnel to work on threat models.

7/2011 – 9/2017 Robert Hurlbut Consulting Services, Enfield, CT
Independent Software Consultant

- Worked on many projects below on a 1099 basis, **each marked with a red plus (+)**.
- Worked on many projects below through **Hurlbut Consulting, Inc.** from 2004 – 2011, **each marked with a red plus (*)**.

+ 07/2011 – 9/2017 Software Consulting Services, Various Clients
Software Security Consultant

- Financial web-based company – secure code reviews, security training (2014-present)
- Finance / Payroll web-based company – secure code reviews, threat modeling (2015 – present)

+ 7/2015 – Current IOActive, Seattle, WA
Software Security Consultant

- Performing threat modeling sessions for client in Silicon Valley, CA.
- Client has mixture of server applications, mobile applications and serves a very large customer base.

***+ 12/2010 – 6/2013, 4/2014 – 9/2015 McKesson Health Solutions, Newton, MA**
Consultant/Team Lead

- Microsoft .NET 4.0 Framework, Windows Presentation Foundation (WPF), Telerik Controls for WPF, SQL Server 2005/2008, Oracle 11g, Windows Communication Foundation (WCF), NHibernate 3.2, FluentNHibernate, Behavior Driven Design (BDD), Test Driven Development (TDD), White, WiX
- Worked within the SCRUM and Agile methodologies as established for the team. Served as a User Interface (UI) Lead Developer on a team using SCRUM (daily stand up meetings, 3 week sprints, stories, etc.).
- Worked on leading a development team using WPF 4.0 for a WPF/XBAP browser application for a care management application. The project was to rewrite an existing older client-server application into a WPF/XBAP application using WPF Web Services.
- Worked with team and management in getting fundamental software practices in place such as Continuous Integration (using TeamCity), BDD and TDD (using StoryQ), and White for white-box testing.
- Worked on strategies for deployment of the WPF/XBAP application and WCF Web Services using Click Once and standard installation practices.
- Worked on improving WCF Web Services performance and NHibernate mapping and performance on the back-end services.
- Worked on back-end integration of health care software products using EDI and Web Services/REST. Created a windows service to manage the integrations for multiple customers.
- Worked with WiX to create install programs for deployment to customers.
- Worked on a WPF project for a customer using XAML, NHibernate.

9/2013 – 9/2014 RSA Security, Inc., Bedford, MA
Software Architect/Consultant

- Microsoft .NET 4.5 Framework, Visual Studio 2013, ASP.NET MVC 4.5, Windows Communication Foundation (WCF) 4.5, Team Foundation Server (TFS) 2013, Test Driven Development (TDD), WiX.
- Provided architecture assistance and mentoring for several projects, including developing a new SDLC process as a way to introduce iterative and agile methodologies for new projects.
- Primary architect and developer on an existing internal/external MVC Web project that used WCF Web Services to interface with an SAP system.
- Provided security code review and improvements to several projects.
- Wrote several data integration tools used for migrating from an older system to a new system.

- Worked to create an updated build, testing, and continuous integration system using TFS 2013.
- Worked to introduce WiX to create install programs for automatic deployments.
- Made improvements to existing secure certificate architecture for existing WCF projects.
- Helped with installation and deployment of secure certificates for several products.

10/2004-6/2011 Hurlbut Consulting, Inc., Agawam, MA

President/Owner/Software Architect/Consultant

- Worked on many projects below on a Corp-to-Corp basis or 1099 basis, **each marked with a red star (*)**.

*** 6/2013 – 9/2013 Mobiquity, Inc., Waltham, MA**

Software Architect/Developer/ Consultant

- Microsoft .NET 4.5 Framework, ASP.NET MVC 4.5, WebAPI 4.5, Visual Studio 2012, Javascript, Mobile tools, Amazon Web Services (AWS) EC2 server.
- Worked on creating a mobile web application using ASP.NET MVC 4.5 and WebAPI 4.5.
- Provided architecture and other assistance to the team.
- Provided security code reviews and improvements for the team.

*** 2/2010 – 11/2010 Cengage Learning, Inc., Boston, MA**

Software Architect/Consultant

- Microsoft .NET 4.0 Framework, SQL Server 2005/2008, SOA using Windows Communication Foundation (WCF), ASP.NET 4.0, Ajax, jQuery, Windows Presentation Foundation (WPF), Microsoft Office 2010, Office OpenXML Toolkit
- Worked within the SCRUM and Agile methodologies as established by the company. Served as a developer on several teams using SCRUM (daily stand up meetings, 3 week sprints, stories, etc.).
- Worked on an Microsoft Office project grading engine using WCF Services, MVC services, and Office OpenXML Toolkit (used automation for Microsoft Access) to grade Microsoft Office student projects and WPF for an instruction builder application that creates the test criteria for grading the projects.

12/2009-4/2010 Verizon Business, Inc., Ashburn, VA

Application Security Consultant

- Microsoft .NET 2.0, 3.5, Java, Ounce Labs - Security Analyst
- Performed security code and PCI reviews for major client (large nationwide retailer)

***+ 10/2009-5/2013 GoGreen EBilling, Inc., Denver, CO**

Software Architect/Developer/Consultant

- Microsoft .NET 3.5 SP1, 4.0, **ASP.NET 4.0**, SQL Server 2005/2008, MVC Pattern, JQuery, JavaScript, AJAX, HTML/CSS, NHibernate / FluentNHibernate / NHibernate.Linq, SOA using Windows Communication Foundation (WCF), Subversion, Cruise Control
- Maintained and enhanced a financial payment web site.
- Introduced Subversion, Cruise Control, Test Driven Development (TDD), and other modern software practices to improve development and maintenance.
- Added ability for political campaigns, non-profit organizations, and other entities to accept financial contributions through the web site.
- Designed and developed highly transactional WCF Services for payment gateway transactions.

*** 11/2008 - 10/2009 Specialized Technology Resources (STR), Inc., Enfield, CT**

Software Architect/Consultant

- Microsoft .NET 3.5 SP1, SQL Server 2005, SOA using Windows Communication Foundation (WCF), Windows Workflow, NHibernate / FluentNHibernate / NHibernate.Linq, ASP.NET 3.5, Ajax, jQuery, ComponentArt ASP.NET Controls, AutoMapper
- Led a programming team as a software architect under a project manager for a new worldwide master file data system.
- Utilized Microsoft MSF, full SDLC, and Agile Methodologies

- Introduced Agile Estimation, Test Driven Development (TDD) and Domain Driven Design (DDD) to the team to efficiently design the new application and begin development. Facilitated stand-up meetings, 2-week iterations, and other Agile techniques for managing the team and code deliverables.

*** 10/2008 Intuit, Inc., Cleveland, OH**

NHibernate and Security Consultant

- Microsoft .NET 2.0, 3.5, SQL Server 2005, SOA using Windows Communication Foundation (WCF), Windows Presentation Foundation (WPF), NHibernate
- Was called in as an NHibernate and software security expert to troubleshoot performance and security issues with an existing software product

*** 11/2007-10/2008 VCI Solutions, Inc., Springfield, MA**

Software Architect/Consultant

- Microsoft .NET 2.0, 3.5, SQL Server 2005, SOA using Windows Communication Foundation (WCF), ASP.NET 2.0/3.5, Telerik ASP.NET Controls, Windows Presentation Foundation (WPF), NHibernate
- Was software architect/programming lead for a new product re-write. The product, Innovation, is a comprehensive media revenue software suite that provides dynamic sales, traffic, and accounting functionality and helps television stations manage the core functions of broadcast and cable operations.
- Introduced Agile Estimation, Test Driven Development (TDD) and Domain Driven Design (DDD) to the team to efficiently design the new application and begin development

*** 08/2007-10/2007 Thomson Financial, Boston, MA**

Consultant

- Microsoft .NET 2.0, SQL Server 2005, ASP.NET 2.0, Web Services, C++
- Added web services support to existing web application to retrieve data

*** 07/2006-07/2007 National Evaluation Systems, Inc., Hadley, MA**

Software Architect/Consultant

- Microsoft .NET 2.0, 3.0, SQL Server 2005, SOA using Windows Communication Foundation (WCF), ASP.NET 2.0
- Taught a two-day class in May, 2006 on “Microsoft .NET and Object Oriented Programming”
- Brought on staff to make software architecture recommendations – introduced Test Driven Development (TDD), Domain Driven Design (DDD), secure programming techniques, and ways to protect sensitive data in SQL Server 2005

*** 11/2006 Gerson Lehrman Group, Inc., New York, NY**

Security Consultant

- Microsoft .NET 2.0, SQL Server 2005, ASP.NET 2.0, Authorization Manager (AzMan)
- Taught 2-day class “Security for the .NET Developer” which covered threat modeling, secure code review, ASP.NET security, database security, Code Access Security, Authorization Manager (AzMan) security, etc.

*** 9/2006 PetEdge, Inc., Peabody, MA**

Security Consultant

- SQL Server 2000, SQL Server 2005
- Performed extensive database security audit of the company’s databases (all SQL Server based)

*** 04/2006-07/2006 Investment Technology Group, Inc., Boston, MA**

Software Architect/Consultant

- Microsoft .NET 1.1, SQL Server 2000
- Performed updates to existing custom code for a CMS application

*** 04/2006 Nexus Energy Software, Inc., Wellesley, MA**

Database Consultant

- Microsoft .NET 1.1, SQL Server 2000

- Analyzed SQL Server 2000 for performance issues, security issues, and other bottlenecks.

*** 03/2006 Algorithmics, Inc., Mount Laurel, NJ**

Security and Database Consultant

- Microsoft .NET 2.0, WinFX 3.0 Beta, WCF, SQL Server 2005, Oracle 10g R2, CodeSmith
- Performed security code audit of the client's smart client application using ClickOnce, WCF services, and SQL Server 2005

*** 12/2005 Telligent Systems, Inc., Dallas, TX**

Security Consultant

- Microsoft .NET 1.1, SQL Server 2000, CodeSmith
- Performed security code audit and penetration testing for one of the client's web sites
- Documented security bugs and provided recommendations for the client

*** 11/2005 – 12/2005 Allscripts, LLC, Chicago, IL**

Security Consultant

- Microsoft .NET 1.1, MESA syslog tools, OpenSSL, SSL
- Developed samples and provided direction for creating secure web services using SSL and other tools
- Developed a syslog .NET client and library to be used for creating medical-based audit messages

*** 12/2004 – 3/2008 iGlobal Technologies, LLC, Cranston, RI**

Computer Consultant/Architect

- Microsoft .NET 1.1/2.0, Visual C++ 7.1, SQL Server 2000, C#/.NET, IIS 6.0, Security, Java 1.5, Tomcat 5.5, Oracle 9i, Macromedia Flash MX 2004, Macromedia Director MX 2004
- Working as Senior Architect/Developer and Project Manager for three sets of teams in developing high volume client/server
- Designed security infrastructure as well as overall physical and software architecture for applications

*** 6/2005 – 10/2007 Salvage Software, Inc., Tucson, AZ and Chicago, IL**

Computer Consultant/Security Consultant

- VS.NET 2003/2005, .NET Framework 1.1/2.0, C#, Installation tools, Access/MSDE/SQL Server 2005 Express, Vista OS development and installation
- Updated original VB 6 and Access application to fix bugs, etc.
- Client contracted my company, Hurlbut Consulting, Inc., to rewrite application in C#/.NET 1.1 using a two-tier client server application approach (i.e. Windows Forms application, central server with a database). Hired another consultant to do most of the development.

*** 7/2003 – 9/2005 Boston Communications Group, Inc., Bedford, MA**

Computer Consultant

- VS.NET 2003, .NET Framework 1.1, ASP.NET, C#, IIS 5/6, Oracle 9i/10g, VB 6.0, COM+, Enterprise Services, .NET Remoting, Web Services, WSE 1.0/2.0, VS.NET Setup and Deployment
- Worked as Senior Developer on project developing a Web-based management system for mobile phones. Phase 1 was written in VB 6.0, COM+, and Oracle 9i. Phase 2 was the previous project re-written as C#, COM+/EnterpriseServices, .NET Remoting, and Oracle 9i.
- Served as Architect for the VB 6.0 version of the above project, and co-Architect for the .NET version of the above project.
- Led **Security Best Practices** seminars for the company to train users/developers on how best to use and develop secure applications within the company.

*** 8/2003 – 11/2005 eDatamate, Inc., Braintree, MA**

Computer/Security Consultant

- Working as Senior Developer/Architect on projects developing Access and Web-based applications. Most applications involved porting older Access applications to SQL Server 2000.

*** 4/2003 – 6/2003 Tier 1 Performance Solutions, Inc., Wilmington, NC**

Computer Consultant

- VS.NET 2003, .NET Framework 1.1, ASP.NET, C#, VB.NET, IIS 5.0, JavaScript, SQL Server 2000, XML, SourceGear Vault
- Working as Developer on a training content management system using C#/VB.NET and ASP.NET. The system is compliant with SCORM training protocols to allow easy integration with other commercial programs.

3/2003 – 12/2003 IT Systems, Inc., Cranston, RI

Computer Consultant/Architect

- VS.NET 2002, .NET Framework 1.0, ASP.NET, VB.NET, IIS 5.0, JavaScript, SQL Server 2000, XML, Visio 2002, Excel
- Working as Senior Architect/Developer on an ASP.NET website for one of the company's clients. The website is a reconfiguration of an old HTML website into the new .NET Framework. The site manages an online catalog of products.

*** 10/2000, 10/2002 – 1/2004 Global Internet Technologies, Cranston, RI**

Computer Consultant/Architect

- Visual Basic 6.0, C#/.NET, IIS 5.0, ASP 2.5, VBScript, JavaScript, ODBC, Access 2000/SQL Server 7.0, SourceSafe 6.0, XML, XSLT, DCOM, Security
- Working as Senior Architect/Developer in a team of two people on the client's site as well as a client of this company.
- Provided security analysis of network and current component systems. Found several areas for improvement, made recommendations, and made changes to improve security across all components and systems.

7/2001 – 10/2002 Geonetics, Inc., Boston, MA

Director of Application Development/Senior Architect

- Visual Interdev 6.0, Visual C++ 6.0, Visual Basic 6.0, Java 1.3, J2EE 1.3, JSP, Java Servlets, ESRI ArcSDE, IIS 5.0, ASP 2.5, VBScript, JavaScript, ADO 2.5, Oracle 8i, SQL Server 2000, SourceSafe 6.0, XML, XSLT, C#, .NET
- Manage/lead/mentor a staff of five developers and two project manager/developers for multiple projects (seven) that are new and/or ongoing at Geonetics.
- Architect/Design/Develop projects for these clients: Massachusetts Highway Department and Central Artery.
- Completed two major projects for Massachusetts Highway involving porting data from old systems to new fiscal management systems. Both projects were completed under budget and on time.

11/2000 - 06/2001 Sprockets.com, Inc., Boston, MA

Computer Consultant

- Visual Interdev 6.0, Visual C++ 6.0, Visual J++ 6.0, IIS 5.0, ASP 2.5, VBScript, JavaScript, ADO 2.5, Oracle 8i, Oracle Objects for OLE, SourceSafe 6.0, XML
- Participated with other C++ and Java developers in a revamping of the main engine for Sprockets.com. The company sells a web collaboration tool for the marketing industry.

4/2000 - 10/2000 Foodline.com, Inc., Boston, MA

Architect/Senior Technical Lead/Computer Consultant

- Visual Interdev 6.0, Visual C++ 6.0, IIS 4.0, ASP 2.0, VBScript, JavaScript, ADO 2.5, SQL Server 7.0, SourceSafe 6.0, XML
- Developer on a small development team (4 people) designing a co-branded sites for the main site. Foodline.com allows customers to make restaurant reservations online.

1/2000 - 3/2000 Centra Software, Inc., Lexington, MA

Computer Consultant

- Visual J++ 6.0, Visual Interdev 6.0, IIS 4.0, MTS 2.0, ASP 2.0, VBScript, JavaScript, ADO 2.1, Site Server 3.0 (Commerce), SQL Server 7.0, ClearCase
- Developer on a small development team (6 people) designing an interface for e-collaboration software marketed by the company. Use of ASP for the front-end work and Java components for the middle/business tiers.

7/1999 - 12/1999 Bain & Company, Inc., Boston, MA

Architect/Computer Consultant

- Visual Basic 6.0, Visual Interdev 6.0, IIS 4.0, MTS 2.0, ASP 2.0, VBScript, JavaScript, ADO 2.1, SQL Server 6.5/7.0
- Project lead/mentor/architect of a small development team designing an information retrieval system for the company Intranet that allows marketing consultants to obtain documents from the company database and from the Internet in one web interface. The Intranet site services 28 offices worldwide.

2/1999 - 6/2001 The Naughton Company, Hingham, MA

Architect/Computer Consultant

- Visual Basic 6.0, Visual C++ 6.0/ATL/COM, Visual Interdev 6.0, IIS 4.0, MTS 2.0, ASP 2.0, VBScript, JavaScript, ADO 2.0, SQL Server 6.5/7.0
- Developing web applications for one of the company's Real Estate sites.

1/1999 - 6/1999 Fidelity Investments, Smithfield, RI

Computer Consultant

- Visual Basic 6.0, Visual C++ 6.0, Oracle 7.3/8.0, InstallShield 5.5, Crystal Reports 6.0
- Designer/developer of COM business objects using MFC/ATL/C++ for retirement fund applications. Prototype user interfaces in Visual Basic.
- Wrote technical and functional specifications for enhancements of retirement applications.

9/1998 - 12/1998 Staples (E-Commerce group) , Westborough, MA

Computer Consultant / Technical Lead

- Visual Basic 5.0, Microsoft Access 97, SQL Server 6.5, NetSoft (AS/400 Interface), InstallShield 5.5, Crystal Reports 4.5/5.0, Active Server Pages, IIS 4.0, Crystal Reports, InstallShield
- Developed/updated applications for order processing through telephone and online connections. Extensive DBA work with SQL Server for projects.

3/1997 - 8/1997, 3/1998 - 9/1998 Medtronic, Inc. , Danvers, MA

Computer Consultant

- Visual Basic 4.0/5.0, Microsoft Access 2.0/97, Visual FoxPro, IBM Client Access (AS/400 Interface), Crystal Reports 4.5/5.0
- Developed interface to AS/400 database servers in Japan to poll inventory data.

9/1997 - 2/1998 Registry of Motor Vehicles, Boston, MA

Computer Consultant

- Visual Basic 5.0, Oracle 7.3, PL/SQL, VS View 2.0
- Technical lead/developer of five-member development team.
- Three-tier client/server application for tracking dealer registration license plates.

9/1996 - 3/1997 Mass Save, Inc., Marlborough, MA

Computer Consultant

- Visual Basic 4.0, Microsoft Access 2.0, SQL Server 6.5, Crystal Reports 4.0
- Completed multi-user applications for updating company database

9/1995 - 9/1996 Boston Voice, Inc., Boston, MA

Computer Programmer

- C/C++ (Borland), Microsoft Access 2.0, Clipper
- Completed projects that enabled automated processing of jobs and reports and to synchronize contact manager databases.

5/1994 - 8/1995 Intelesis, Inc., Columbus, IN

Computer Systems Specialist/Programmer

- Visual Basic 3.0, Microsoft Access 2.0
- Completed software projects that enabled the company to track telecommunications data.

5/1991 - 9/1993 DAC Services, Tulsa, OK

Software Engineer/ Programmer

- C, Paradox
- Completed software projects that enabled voice mail screening.

6/1986 - 5/1991 Southwestern Power Administration, U.S. Department of Energy, Tulsa, OK

Engineering Technician (Student Trainee)

- C, FORTRAN, dBase III+, PRIME mainframe
- Developed engineering applications.

INDEPENDENT RESEARCH, SPEAKING ENGAGEMENTS, & WRITING

- Co-wrote Threat Modeling Manifesto with 14 other industry experts on Threat Modeling in 2020 (<https://threatmodelingmanifesto.org>)
- Wrote series of articles on *Security Best Practices* for code and database development for .NET community.
- Wrote article on *SQL Injection for .NET Developers Journal*.
- Frequently wrote technical content on web blog (<https://roberthurlbut.com/blog>).
- Speaker at user groups and national / international conferences on security (Application Security / Threat Modeling), .NET, and developer topics.
- Wrote courses for NHibernate training, ASP.NET MVC 2.0 training, OWASP security courses for LearnNowOnline.
- Manage two local user groups, one on architecture (since 2004), and one on security (since 2016).
- Co-hosted application security podcast.

EDUCATION

Master of Science, Cyber Security, Southern New Hampshire University, Manchester, NH, 2020

Bachelor of Science, Mathematics, Oklahoma State University, Stillwater, OK, 1990

CERTIFICATION

(ISC)2 Certified Secure Software Lifecycle Professional (CSSLP), 2014 - 2023

MANAGEMENT & MENTORING COMPETENCIES

Lead Developer / Architect / Mentor / Project Manager / Director

- Communication & organizational skills
- Mentoring & encouraging best work from team members
- Creating work schedules & weekly staff meetings
- Ongoing consultation
- Understanding business challenges
- Applying technical experience to develop team-based solutions

EXHIBIT B

EXHIBIT B

10,713,366 “Systems and methods for automated threat model generation from third party diagram files”	The '366 Patent is Anticipated by US 11,200,228 (“Keenan”)
1[p]. A threat modeling method, comprising:	Keenan discloses this claim limitation. For example, Keenan discloses: “A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.” Keenan at Abstract. “Embodiments include a standard of diagramming for classifying the objects and defining the format of the diagram. Design of object classifications is suited to capture the properties of the information assets in a way that associates the risk, cost, quality, accountability of the assets to the product chain that produces calcs which the business utilizes to set premiums and reserves, manage investment risk and produce financial reports and compliance reports. The same end-to-end production chain of information assets that produces calcs is represented in the topological repository and connected with its respective properties using the IOE. These properties (e.g. costs, risks and accountability) are then compiled and aggregated via their objects in the topology along the production chain which produced the calc and as such can then be associated with the business outcome that the calc enabled.” Keenan at 1:31-46. “The system stores, reveals and links characteristics (Structural Metadata) of each object in the topology either within the IOE or in a linked repository, such that accountability, benefits and consequences of using those assets can easily and repeatedly be measured in the cost and profitability of the outcomes those assets enable. The system provides a wide variety of visual presentation options of the topology allowing the user to view , using GUI , a subset of the topology by selecting use , object function class , functional specificity , relationship to specific object and other criteria . It provides a medium for applying user - provided logic to the topology directly , so that any component

	or aspect of it can be isolated , retrieved , and assessed.” Keenan at 1:47-58. “d) Diagrams are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams are not stored in the IOE; instead the diagram serves as a communication medium to and from users and the IOE. e) There is a one to one relationship between an object class and a diagram shape. f) The diagramming standards are pre-specified sets of object shapes and formats and other visual elements. g) Diagrams created in the IOE automatically synchronize with same intelligence stored in structural metadata associated with the object; objects are related to other objects through “links” that obey pre-set rules, thus establishing the logic of the topological structure and enforcing integrity within the system. h) The IOE preferably differentiates between ‘hard data’ (i.e. observations and measurements, and arithmetic combinations thereof) and ‘calculations’ which represent modeled combinations of inputs. The former are referred to herein only as ‘data’ and the latter as ‘calcs’.” Keenan at 2:3-23. “MLD (Model Level Diagram): Type of IOE diagram centered around a specific model. The purpose of this diagram is to depict feeds into the model from all sources (including other models) and the destination of the output (including other models). MLDs are joined together to form Stream Diagrams.” Keenan at 4:46-51. “Structural Metadata: For this document, the term “Characteristic” is used interchangeably. Examples of structural metadata for most objects: owner, risk measurement, age, business affiliation, users” Keenan at 4:64-67. “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise’s analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is
--	--

	provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “Model Objects receive data and/or outputs from other models, and generate outputs restricted to: i) Forecasts/projection; ii) Estimates of unknown/unobservable values; iii) Probability measures including probability distributions and values derived therefrom; and iv) Quantitative hypothesis test values. It is the generation of any of these outputs that defines a model. Models have two modes that are separately/simultaneously captured within the topology: development mode, and deployment mode. Data and other inputs used to develop the model (including data storage and data flow objects) may differ from those used to run the model once deployed. Moreover, the development platform may differ from the deployment platform. For some models, one mode may not exist or not be identifiable. This could include cases where a model exists for extended periods of time in development mode without being deployed, as well as vendor models where the development mode information is limited or not provided. However, the ‘modal’ nature of models must an inherent characteristic of the object class.” Keenan at 7:41-59. ‘The presentation of a topology's objects in the GUI is preferably designed to maximize the amount of information depicted in a sub-set of the topology (referred to as a “View”) by using particular shapes and lines to depict the value, range of values of a specific characteristic or the sub-class of the object. The shapes themselves are also used as dimensions as they differentiate object classes. Aside from shapes, which represent the object class of the object, there are at least two other dimensions for depicting intelligence; Color and Line Weight. The two dimensions of Color and Line Weight are preferably applied to visual elements of the Topology and are preset rather than being a user option. The system may permit user-defined dimensions in the case of special Views. For instance, a shape fill color may represent platform association in the default view, but represent model risk in a specific Palette View (see below). The number of characteristic or sub-class values possible for each visual component is thus limited only by the differences the normal human eye can detect on today's screens” Keenan at 13:4-22. “Without an upstream limit, the diagram shows all upstream relationships.) The GUI preferably permits user selection of
--	---

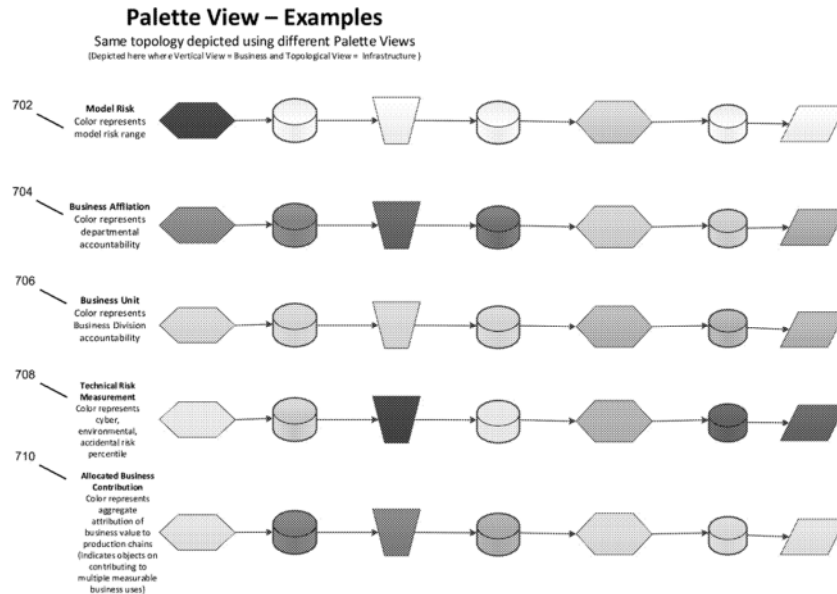
other options to view the topology, which can be selected in combination with one another. A summary of View options, in accordance with an embodiment, is shown in Table 3:” Keenan at 14:7-68; 15:1-10.

TABLE 3

View Name	Options	Description
OBJECT CLASS	All Classes	All object classes are depicted
	Specify Object Classes (Flow objects will not be shown unless all object classes are selected)	User can select one or more classes to depict. Data Flow Lines will not be depicted unless all classes are selected. In lieu of data flow lines, objects will be joined by lines that look like data flow lines but the arrow heads will be a different shape.
HORIZONTAL	Linear	An instance of a particular Data Storage object appears for every extract and load downstream of the initial instance of the Data Storage object in the View
	Network	all Data Storage objects appear only once in the View
VERTICAL	Business	Highest level functionality
	IT Architecture	Application centric
	IT Infrastructure	Hardware centric
PALETTE	Platform	color fills the object shapes based on the platform that hosts the object. Objects not hosted by platforms will have white fill.
	Model Risk	color fills the object shapes based on the value of the risk characteristics for the models in the view
	Data Quality	color fills the Flow arrowhead shapes based on the value of the value of data quality characteristic for the data elements feeding the object. This palette can be selected
	Risk Capital Allocation	color fills the object shapes to indicate the absolute value of Risk capital allocated to the object.
	Business Entity	color fills the object shapes based on the Business Entity or Entities that use the output from a Path of objects
	Business Owner	color fills the object shapes based on the business entity that owns or co-owns the object
	Value by Business Metric	color fills the object shapes to differentiate the value of the business metric(s) associated with each production chain of objects in the view.
	Use	color fills the object shapes based on the Uses within the View
	Age	color fills the object shapes based on the value of the age characteristic for that object
	Technical Threat Vector	color fills the object shapes based on the value fo the technical risk ranking characteristic of that object
	Operational Risk	color fills the object shapes based on the value of operational risk characteristic of the object or its associated control
TABLE 3 (continued)		
View Name	Options	Description
ARROW-HEAD	None	Arrowheads depicted in solid black
	Data Quality	color fills the Flow arrowhead shapes based on the value of the value of data quality characteristic for the data elements feeding the object. This palette can be selected

Keenan at Table 3

	“An example of a Palette View, in accordance with embodiments of the systems, are shown in FIG. 7. Palette View options allow the user to highlight a particular set of Characteristics that are preset such as risk, business ownership, age, etc. A Palette is a View that embeds the intelligent of a different set of key characteristics other than the default View. This is accomplished by switching the mapping of the Visual Elements to characteristics different than what is in the default View. Examples of Palette Views include: “Model Risk color fills the object shapes based on the value of the risk characteristics for the models in the view ... “Technical Threat Vector: color fills the object shapes based on the value of the technical risk ranking characteristic of that object ... Operational Risk: color fills the object shapes based on the value of operational risk characteristic of the object or its associated control” Keenan at 16:1-29. “In the example of FIG. 7, a Model Risk view 702 uses color to represent a range of risk based on the risk characteristics for models in the view. A Business Affiliation view 704 uses color to represent differences in accountability for different business departments. A Business Unit view 706 uses color to represent business division accountability. A Technical Risk Measurement view 708 uses color to represent different types of risk (e.g., cyber, environmental, accidental, etc.). An Allocated Business Contribution view 710 uses color to represent aggregate attribution of business value to production chains (e.g., to indicate objects contributing to multiple business uses).” Keenan at 16:40-51.
--	---



FIG

Keenan at Fig. 7.

“An Arrowhead View is also used in embodiments of the present system to depict Data Quality measurement. The method is by color coding the arrowhead fill space. The default Arrowhead color is black. In this View, for arrows showing the flow between objects in the View, the arrowheads fill with color based on the risk measurement of the upstream calc and data at that point. For example, three relative gradations of risk could be represented (e.g., red for the highest 10% of risk scores; yellow for the upper half of risk scores; and green for the lowest half of risk scores. A clear arrow can represent that data is missing.) Values to determine Data Quality score will be set by Contributors.” Keenan at 16:52-63.

“The GUI also preferably includes an option for calculating and displaying the Riskiest Path. The user selects one of the risk metrics and the IOE computes based on pre-loaded formulae the path with the highest risk based on the metric selected by the user. The riskiest path will appear highlighted (e.g., in red) as seen in FIG. 11. A sample list of risk metrics include: Model Risk without exposure; Model Risk with exposure; Data Quality; Ops Risk; Tech Risk; and Aggregated Risk.” Keenan at 17:32-40.

“Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction

	of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View; Depict Riskiest Path, which depicts the Path whose objects in aggregate measure riskiest versus all other Paths in the View, using a formula preferably based on Risk value characteristics of the Objects; Depict Lowest Data Quality Path, which depicts the Path whose data elements in aggregate measure lowest in data quality versus all other Paths in the View, using a formula preferably based on Data Quality value characteristics of the Data Elements; Calculate Aggregate risk by View, which depicts risk values of the objects of the View aggregated based on an aggregation formula.” Keenan at 18:6-22. “Changes to characteristics can also include formulas that calculate risk parameters based on new raw values put into to characteristics of one object but flow downstream to other objects. Entire paths are preferably monitored for changes to aggregate characteristics values. For instance, a production chain of four models might be set to alert downstream consumers of those four models if the aggregate model risk of the four models exceeded a certain amount.” Keenan at 20:34-51. Lasso Report – Example Showing the selection via the free form tool of desired Objects (Reported from where vertical view = Business and Topological View = Infrastructure) Select Characteristics of Object Classes Lassoed Processing Objects: (drop down menu) - Owner Name - Production date - Model Name - etc. Model Objects: (drop down menu) - Validation Date - Series - Owner - Business Unit - etc. Storage Objects: (drop down menu) - Owner - Date Viewed - Production date - etc. View Objects: (drop down menu) - Data elements - Data quality - etc. FIGURE 16 Keenan at Fig. 16
1[a]. providing one or more databases, the one or more databases comprising:	Keenan discloses this claim limitation. For example, Keenan discloses: “A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the

topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.” Keenan at Abstract.

“According to embodiments of the present invention, an Integrated Object Environment (IOE) running in a graph database environment is used. Diagrams of the production chain of information assets are created. These diagrams depict data flows linking together these information assets, for which the final product is calculated output (“calcs”) used for a specific business purpose. These diagrams are loaded into a repository that stores, reveals and allows for maintenance of the topology.” Keenan at 1:22-30.

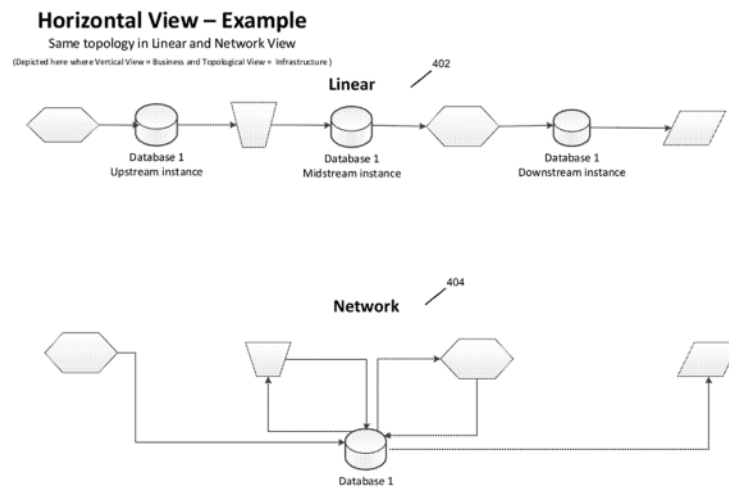


FIGURE 4

Keenan at Fig. 4.

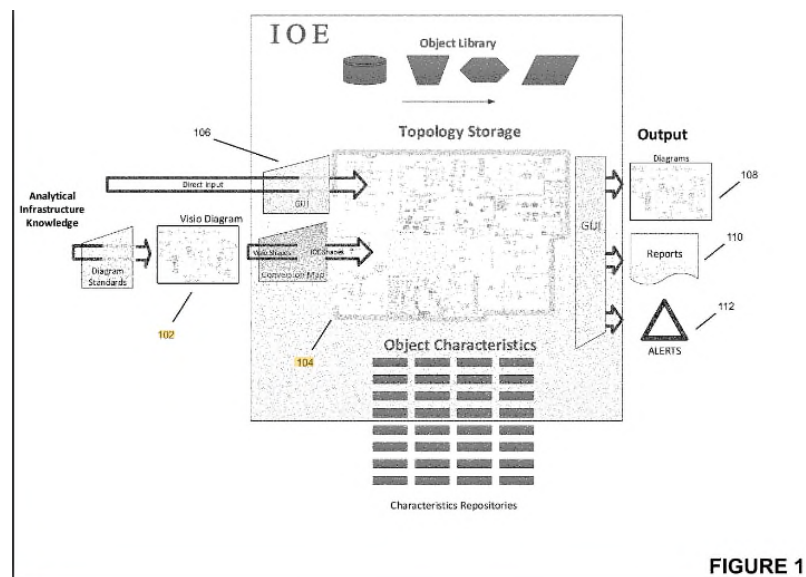
“The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database.

	Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:28-44. “In greater detail with respect to FIG. 1, the IOE preferably resides in a graph database utilizing, object-oriented storage technology, where each physical infrastructure component is represented by an object class from a parsimoniously designed object class set. The relationships among objects are represented by a parsimonious set of pre-defined links. The data flow between the objects is also defined as an object class. The object classes and links have features that allow for its characterization including specification of its state. The relationships characterizing the links between objects are preferably fixed (non-modifiable), to accommodate a restriction on data flow objects feeding into other data flow objects. This environment permits storage of the objects that represent the analytic infrastructure, and the structural metadata. A scripting language is preferably adopted and an interface designed that allows users to build and store/run complex programming statements that apply logic directly to the topology or any subset thereof.” Keenan at 5:45-62. FIGURE 1 Keenan at Fig. 1.
1[b]. a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of	Keenan discloses this claim limitation. For example, Keenan discloses: “A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph

the threats is associated with at least one of the threat model components through the one or more databases;	database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.” Keenan at Abstract. “Embodiments include a standard of diagramming for classifying the objects and defining the format of the diagram. Design of object classifications is suited to capture the properties of the information assets in a way that associates the risk, cost, quality, accountability of the assets to the product chain that produces calcs which the business utilizes to set premiums and reserves, manage investment risk and produce financial reports and compliance reports. The same end-to-end production chain of information assets that produces calcs is represented in the topological repository and connected with its respective properties using the IOE. These properties (e.g. costs, risks and accountability) are then compiled and aggregated via their objects in the topology along the production chain which produced the calc and as such can then be associated with the business outcome that the calc enabled. The system stores, reveals and links characteristics (Structural Metadata) of each object in the topology either within the IOE or in a linked repository, such that accountability, benefits and consequences of using those assets can easily and repeatedly be measured in the cost and profitability of the outcomes those assets enable.” Keenan at 1:31-52. “The system stores, reveals and links characteristics (Structural Metadata) of each object in the topology either within the IOE or in a linked repository, such that accountability, benefits and consequences of using those assets can easily and repeatedly be measured in the cost and profitability of the outcomes those assets enable.” Keenan at 1:47-52. “MLD (Model Level Diagram): Type of IOE diagram centered around a specific model. The purpose of this diagram is to depict feeds into the model from all sources (including other models) and the destination of the output (including other models). MLDs are joined together to form Stream Diagrams.” Keenan at 4:46-51. “Structural Metadata: For this document, the term “Characteristic” is used interchangeably. Examples of structural
--	--

metadata for most objects: owner, risk measurement, age, business affiliation, users” Keenan at 4:64-67.

“The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44.



Keenan at Fig. 1.

“Model Objects receive data and/or outputs from other models, and generate outputs restricted to: i) Forecasts/projection; ii) Estimates of unknown/unobservable values; iii) Probability measures including probability distributions and values derived therefrom; and iv) Quantitative hypothesis test values. It is the generation of any of these outputs that defines a model. Models have two modes that are separately/simultaneously captured within the topology: development mode, and deployment mode.

	Data and other inputs used to develop the model (including data storage and data flow objects) may differ from those used to run the model once deployed. Moreover, the development platform may differ from the deployment platform. For some models, one mode may not exist or not be identifiable. This could include cases where a model exists for extended periods of time in development mode without being deployed, as well as vendor models where the development mode information is limited or not provided. However, the ‘modal’ nature of models must an inherent characteristic of the object class.” Keenan at 7:41-59. ‘The presentation of a topology's objects in the GUI is preferably designed to maximize the amount of information depicted in a sub-set of the topology (referred to as a “View”) by using particular shapes and lines to depict the value, range of values of a specific characteristic or the sub-class of the object. The shapes themselves are also used as dimensions as they differentiate object classes. Aside from shapes, which represent the object class of the object, there are at least two other dimensions for depicting intelligence; Color and Line Weight. The two dimensions of Color and Line Weight are preferably applied to visual elements of the Topology and are preset rather than being a user option. The system may permit user-defined dimensions in the case of special Views. For instance, a shape fill color may represent platform association in the default view, but represent model risk in a specific Palette View (see below). The number of characteristic or sub-class values possible for each visual component is thus limited only by the differences the normal human eye can detect on today's screens” Keenan at 13:4-22. “Model Risk color fills the object shapes based on the value of the risk characteristics for the models in the view” Keenan at 14:35-39. “Technical Threat Vector: color fills the object shapes based on the value of the technical risk ranking characteristic of that object” Keenan at 14:61-63. “An example of a Palette View, in accordance with embodiments of the systems, are shown in FIG. 7. Palette View options allow the user to highlight a particular set of Characteristics that are preset such as risk, business ownership, age, etc. A Palette is a View that embeds the intelligent of a different set of key characteristics other than the default View. This is accomplished by switching the mapping of the Visual
--	--

Elements to characteristics different than what is in the default View. Examples of Palette Views include:

“Model Risk color fills the object shapes based on the value of the risk characteristics for the models in the view

...

“Technical Threat Vector: color fills the object shapes based on the value of the technical risk ranking characteristic of that object

...

Operational Risk: color fills the object shapes based on the value of operational risk characteristic of the object or its associated control” Keenan at 16:1-29.

“In the example of FIG. 7, a Model Risk view 702 uses color to represent a range of risk based on the risk characteristics for models in the view. A Business Affiliation view 704 uses color to represent differences in accountability for different business departments. A Business Unit view 706 uses color to represent business division accountability. A Technical Risk Measurement view 708 uses color to represent different types of risk (e.g., cyber, environmental, accidental, etc.). An Allocated Business Contribution view 710 uses color to represent aggregate attribution of business value to production chains (e.g., to indicate objects contributing to multiple business uses).” Keenan at 16:40-51.

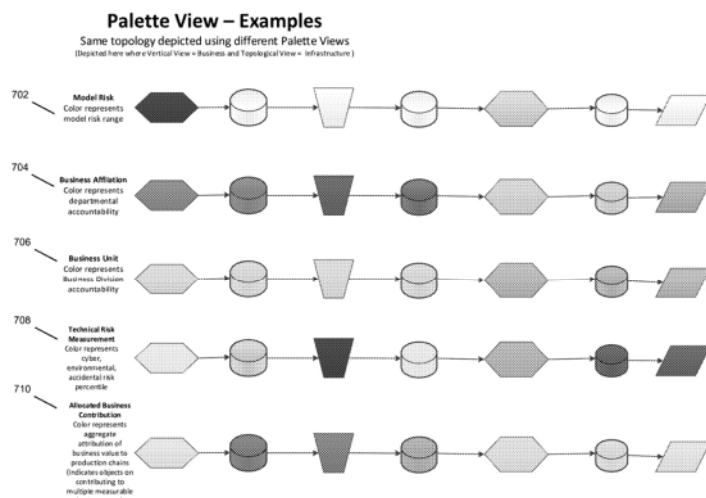


FIGURE 7

Keenan at Fig. 7

“An Arrowhead View is also used in embodiments of the present system to depict Data Quality measurement. The method is by color coding the arrowhead fill space. The default Arrowhead color is black. In this View, for arrows showing the flow

	between objects in the View, the arrowheads fill with color based on the risk measurement of the upstream calc and data at that point. For example, three relative gradations of risk could be represented (e.g., red for the highest 10% of risk scores; yellow for the upper half of risk scores; and green for the lowest half of risk scores. A clear arrow can represent that data is missing.) Values to determine Data Quality score will be set by Contributors.” Keenan at 16:52-63. “The GUI also preferably includes an option for calculating and displaying the Riskiest Path. The user selects one of the risk metrics and the IOE computes based on pre-loaded formulae the path with the highest risk based on the metric selected by the user. The riskiest path will appear highlighted (e.g., in red) as seen in FIG. 11. A sample list of risk metrics include: Model Risk without exposure; Model Risk with exposure; Data Quality; Ops Risk; Tech Risk; and Aggregated Risk.” Keenan at 17:32-40. “Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View; Depict Riskiest Path, which depicts the Path whose objects in aggregate measure riskiest versus all other Paths in the View, using a formula preferably based on Risk value characteristics of the Objects; Depict Lowest Data Quality Path, which depicts the Path whose data elements in aggregate measure lowest in data quality versus all other Paths in the View, using a formula preferably based on Data Quality value characteristics of the Data Elements; Calculate Aggregate risk by View, which depicts risk values of the objects of the View aggregated based on an aggregation formula.” Keenan at 18:6-22. “Changes to characteristics can also include formulas that calculate risk parameters based on new raw values put into to characteristics of one object but flow downstream to other objects. Entire paths are preferably monitored for changes to aggregate characteristics values. For instance, a production chain of four models might be set to alert downstream consumers of those four models if the aggregate model risk of the four models exceeded a certain amount.” Keenan at 20:34-51.
--	--

1[c]. providing one or more mapping files communicatively coupled with the one or more databases,	Keenan discloses this claim limitation. For example, Keenan discloses: “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “The screen also contains a mini map 204, located in the upper left corner, displaying a depiction of the topology. The default view of the mini map 204 is preferably of the entire company's topology. Objects and labels may be too small to be seen individually, but the mini map 204 nonetheless shows the user where within the entire company topology the current View is located.” Keenan at 11:65-12:4. “Within the mini map 204 is a locator box 206. Regardless of how the user zooms in and out of the Mini Map, the Locator Box encompasses the same topology and so expands or contracts to remain in sync with the mini map display 204 as the zoom function is exercised. When the cursor is not in the Mini Map 204, the Locator Box 206 displays the same boundaries represented by the View in the View Box 202. This is accomplished by automatically shifting position, shrinking and expanding borders, to encompass topology represented in the View Box 202. The functionality of the mini map 204 is more fully described with respect to Table 2.” Keenan at 12:11-23.
--	--

TABLE 2		
Cursor position	On Mini Map outside Locator box	Inside Locator Box
Single left click	Demarcates the Mini Map based on Landmarks set up by the IOE System Administrator. This is a toggle on/off (a second left click removes Landmarks)	Activates locator box boundary allowing user to expand or contract any side or corner of the locator box
Double left click	opens the Mini Map to full screen overtop existing screen. Escape key or double left click shrinks Mini Map back to normal size. Mini Map and Locator Box retain full functionality in this full screen mode.	Transfers topology inside the Locator Box to replace what is in the View Box
Single right click	Brings up properties of the Mini Map. One property allows the user to change the default View options of the Mini Map	Allows user to open another View Box filled with the contents of the Locator Box. Up to six View Boxes can be open at the same time.
Mouse wheel forward	Zooms in the Mini map. The Locator Box boundaries remain fixed relative to objects rather than remaining fixed relative to size on the screen.	Zooms in the Mini map. The Locator Box boundaries remain fixed relative to objects rather than remaining fixed relative to size on the screen.
Mouse wheel back	Zooms out the Mini map. The Locator Box boundaries remain fixed relative to objects rather than remaining fixed relative to size on the screen.	Zooms out the Mini map. The Locator Box boundaries remain fixed relative to objects rather than remaining fixed relative to size on the screen.

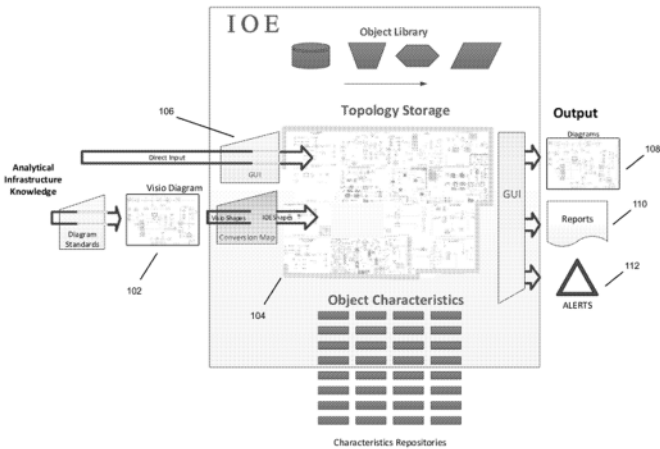
Keenan at 12:25-62.

“A Palette is a View that embeds the intelligent of a different set of key characteristics other than the default View. This is accomplished by switching the mapping of the Visual Elements to characteristics different than what is in the default View. Examples of Palette Views include: Model Risk: color fills the object shapes based on the value of the risk characteristics for the models in the view.” Keenan at 16:5-11.

“The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function.” Keenan at 18:23-28.

	“A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match.” Keenan at 18:36-40. Palette View – Examples Same topology depicted using different Palette Views (Displayed here where Vertical View = Business and Topological View = Infrastructure) FIGURE 7 Keenan at Fig. 7
1[d]. the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”); and	Keenan discloses this claim limitation. For example, Keenan discloses: “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44.

	“Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View; Depict Riskiest Path, which depicts the Path whose objects in aggregate measure riskiest versus all other Paths in the View, using a formula preferably based on Risk value characteristics of the Objects; Depict Lowest Data Quality Path, which depicts the Path whose data elements in aggregate measure lowest in data quality versus all other Paths in the View, using a formula preferably based on Data Quality value characteristics of the Data Elements; Calculate Aggregate risk by View, which depicts risk values of the objects of the View aggregated based on an aggregation formula. The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function. Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams. A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:6-50. “Turning to FIGS. 14 and 15, a compare function of the IOE is described. A user selects two diagrams: a diagram previously imported into Visio and a diagram to be imported. One will be
--	---

	the “BASE” and the other will be the “CHANGED”. Comparisons can be done from any View Option combination. Embodiments preferably include two different methods for comparing; 1) Melding, where all objects from both diagrams are displayed and color coded to indicate which are Common, which are New and which were on the Base diagram only (FIG. 14); and 2) Twin Screen, where both the diagrams are displayed with the same color code representing the change in Object from Base to Changed (FIG. 15).” Keenan at 19:17-29. “Characteristics from Structural Metadata Repositories for the respective object in the IOE are preferably loaded into the IOE on a daily basis. History is preferably saved for a defined period. The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. The IOE preferably allows queries originating in Structural Metadata repositories to query the topology in the IOE.” Keenan at 21:10-17.  FIGURE 1 Keenan at Fig. 1.
1[e]. in response to receiving one or more user inputs, using one or more user interfaces displayed on one or more computing devices communicatively coupled with the one or more databases:	Keenan discloses this claim limitation. For example, Keenan discloses: “The system provides a wide-variety of visual presentation options of the topology allowing the user to view, using GUI, a subset of the topology by selecting use, object function class, functional specificity, relationship to specific object and other criteria. It provides a medium for applying user-provided logic to the topology directly, so that any component or aspect of it can be isolated, retrieved, and assessed.” Keenan at 1:53-60

	“d) Diagrams are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams are not stored in the IOE; instead the diagram serves as a communication medium to and from users and the IOE.” Keenan at 2:3-6 “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match.” Keenan at 18:36-40. “FIG. 3 is a diagram illustrating drop down menu options for a view portion of a graphical user interface for use in an integrated object environment, in accordance with an embodiment of the invention;” Keenan at 2:51-54. “Diagram: Communication vehicle between user and the IOE and between the IOE and the user depicting static topological information in a format selected by, and consumable by users which can be set in documentation, stored as a file in other applications. This term contrasts with the term for the visual depiction of topology seen in the GUI of the IOE which is referred to as a “View”. Another way to understand a diagram is as a fixed portion of the topology. A diagram is topology packaged to be communicated.” Keenan at 4:5-14 “View: Depiction of topology in the IOE via GUI. A view that is exported or printed from the IOE is a “Diagram”. View Box: that portion of the IOE GUI screen that contains the View.” Keenan at 5:24-27 “Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:40-44 “Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View” Keenan at 18:6-13.
--	--

	“In embodiments of the present system, there are preferably four modes for the user to interact with the IOE via a Graphical User Interface (GUI): 1) a View mode, where the user can view any section of the topology and Characteristics by Object; 2) a Modify mode, where the user can add, change and delete Objects or Characteristics; 3) a Compare mode, where the user can visualize the difference between two diagrams; and 4) a Query mode, where the user can interrogate the topology and Characteristics using standard queries and user-generated queries (script writing). Query results can be tabular (reports) or diagrammatic results in which case the diagram will appear in the View Box. Each mode brings up a different screen format. The portion of the GUI that the user sees is referred to herein as the “screen” as opposed to the logic which dictates what appears on the screen which would be the unseen parts of the GUI.” Keenan at 11:42-57. “Turning to FIGS. 2 and 3, an exemplary GUI for a View mode is now described, in accordance with an embodiment. The screen contains a view box 202 that occupies most of the screen. Within the view box the user is presented with the topology to view, modify or compare. The view box is view-only in the View and Compare modes, while it is interactive in the Modify mode.” Keenan at 11:58-64.
--	--

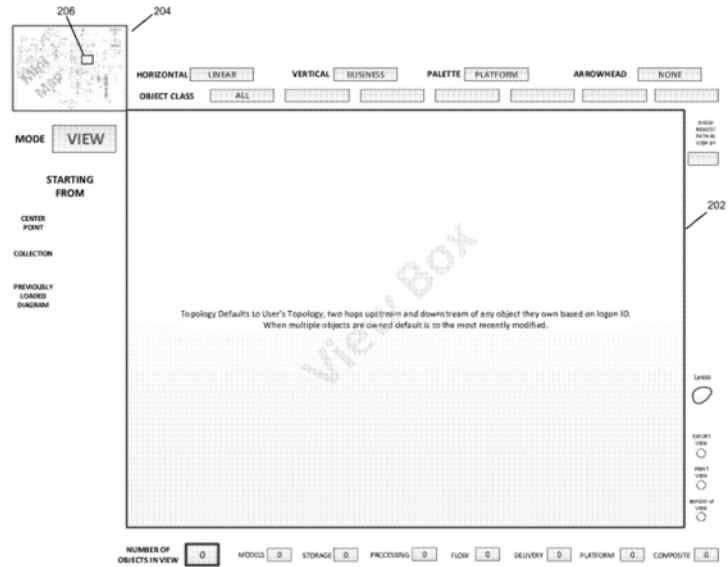


FIGURE 2

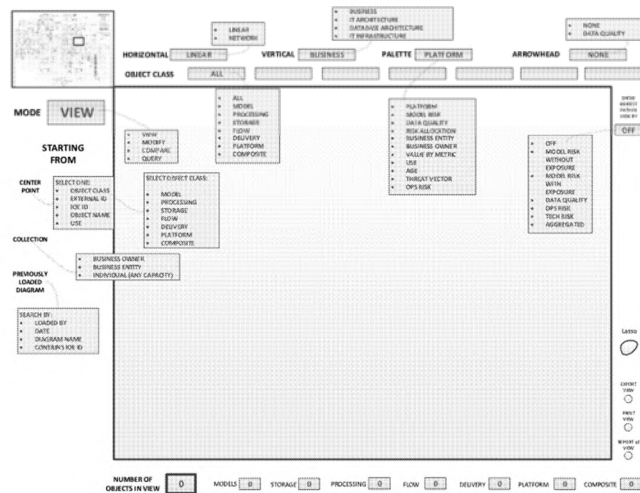


FIGURE 3

Keenan at Figs. 2, 3.

“The GUI of the IOE in embodiments of the present system include multiple ways for a user to select a subset of the topology to view, for example: Mini Map; Single Object Selection; Multiple Object Selection; and other options. With a Mini Map, the user selects the View by entering one or more of the following from a drop-down menu (the list below is not an exhaustive list). The scope of the topology presented includes all objects connected upstream and downstream within the selection criteria requested. The GUI also provides the user with an option to have the output appear in a table which can be exported as excel file. If the criteria entered is not currently stored in the IOE, the user is given the option to add new topology, described

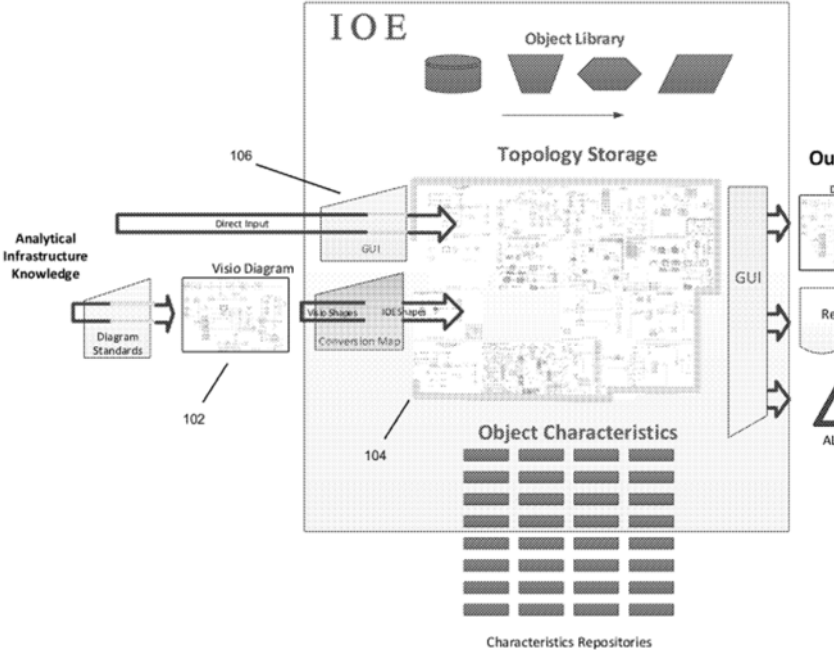
	more fully below. With a Single Object Selection, the user may specify a Use, ObjectID and/or Object Name to identify an object, along with all objects feeding into and all objects being fed by the Object selected. With a Multiple Object Selection, the GUI permits the user to specify a Business entity, a diagram previously imported, a person's name, and/or an object class or subclass. Once the selection is made, the GUI permits the user to refine the view by filtering on a variety of criteria. For example, filters may include limiting to upstream or downstream paths only. Another filter may permit a user to enter a start point or end point (or both) in addition to the View criteria above (e.g., End point=Technical Pricing Model, showing all objects related to business unit=Commercial. The resulting diagram is drawn showing all objects connected to any object which feeds ultimately to the Technical Pricing Model. Without an upstream limit, the diagram shows all upstream relationships.) The GUI preferably permits user selection of other options to view the topology, which can be selected in combination with one another.” 13:45-14:10. “It is understandable to those skilled in the art that all or a part of steps of the processes in the preceding embodiments is preferably implemented by relevant computing hardware instructed by a program. The program may be stored in a computer readable storage medium. The storage medium may include a ROM, a RAM, a magnetic disk or a compact disk.” 21:27-33. Keenan dedicates a substantial portion <i>See also</i> Keenan at 12:64-14:10; 18:36-50 and Figs. 4-16.
1[f] using the one or more computing devices, reading a data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;	Keenan discloses this claim limitation. For example, Keenan discloses: “According to embodiments of the present invention, an Integrated Object Environment (IOE) running in a graph database environment is used. Diagrams of the production chain of information assets are created. These diagrams depict data flows linking together these information assets, for which the final product is calculated output (“calcs”) used for a specific business purpose. These diagrams are loaded into a repository that stores, reveals and allows for maintenance of the topology.” Keenan at 1:24-30 “Junction: Multiple Paths join in at a single object and multiple Paths exit from this object.

	Junction Problem: Situation where multiple Paths join at a Junction but no information from one or more upstream Paths is conveyed, in any form, by the downstream Paths. This creates a problem for tracking the data lineage and requires that data elements be characteristics of Data Flow objects in order to trace the flow of information through a Junction.” Keenan at 4:39-45 “Module: internal component of a model which produces output consumed within the model. Path: A sequential set of objects connected by data flow without a fork. It is defined by a single object on the upstream end and a single object on the downstream end. It is possible to have a multiple paths starting at the designated upstream object and ending at the designated downstream object although this would indicate a probable misconception of the diagram or process.” Keenan at 4:54-60 “In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure.” Keenan at 5:33-40 “The data flow between the objects is also defined as an object class. The object classes and links have features that allow for its characterization including specification of its state. The relationships characterizing the links between objects are preferably fixed (non-modifiable), to accommodate a restriction on data flow objects feeding into other data flow objects. This environment permits storage of the objects that represent the analytic infrastructure, and the structural metadata. A scripting language is preferably adopted and an interface designed that allows users to build and store/run complex programming statements that apply logic directly to the topology or any subset thereof.” Keenan at 5:51-62 “While data flow objects have direction, with the source and destination information represented in the topology, direction is implicitly included and so an explicit direction feature does not need to be embedded in the objects themselves. Data element level information preferably identifies whether it is “datum” or is “calc”. Data flow objects link to downstream objects using a “passes to” concept in the data flow object, as opposed to
--	--

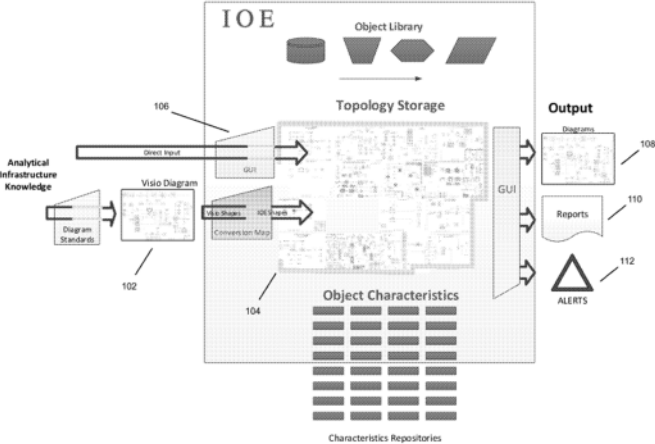
	characterizing the linkage as “receives from” in the downstream object.” Keenan at 7:25-34 “Composite Objects are composites of multiple objects designed to accommodate multiple class characteristics. For example, a spreadsheet may behave as both a data storage object and as a model. Essentially, the composite object acts as a wrapper, providing the necessary information to keep the underlying objects fully and independently identified, for each object class it is a member of, and providing an association between class characteristics and the components of the composite object. That is, to re-use the multi-function spreadsheet example above, the composite object would be a dual list of defining characteristics for ‘models’ and ‘data source’ objects, and provide a method by which data flow objects could be separately linked to the components. More specifically, some data may flow into the object only to be stored and later fed into some downstream objects and that data flow object needs to exist independently and be fully identified. At the same time, another data flow object may flow into the composite purely to feed the model never to be re-directed to any other objects, and that data flow object needs to exist independently and be fully identified.” Keenan at 8:21-41 “Turning to FIG. 4, a Horizontal View is described, in accordance with an embodiment. The GUI permits the Horizontal View to be shown in multiple ways. For example, a Linear View 402 depicts the production chain of information assets as the data flow sequentially progresses downstream through the chain of information assets. This Linear View is created by following the path of data elements in the data flow objects as they are passed from object to object. Since objects are stored in the IOE as a single instance, the multiple loads and extracts to the same database in a View are depicted as unclassified objects in the topological space allowing depiction of the proper sequence of the production chain. These Unclassified Objects are created dynamically whenever a Linear View is requested. The Characteristics of these dynamically created Unclassified Objects equals the Characteristics of the Storage Object for which it represents an instance of. For example, if model X produces calc “A” and calc “A” is subsequently loaded to database STS, and model Y extracts calc “A” from STS, then by inference, model X and model Y are on the same production chain and X is upstream of Y. On the Linear View, Model X will be connected to Model Y with a Data Flow object to a Unclassified Object representing the
--	--

	storage in STS followed by a Data Flow object connecting the Unclassified Object to Model Y. If STS is loaded and extracted from multiple locations along the production chain, then a proxy shape will be depicted on the View at each load and extract point to maintain production chain continuity.” Keenan at 15:15-43 “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function.” 18:23-28. “Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.” Keenan at 18:29-39. “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing
--	---

	Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50. “Turning to FIGS. 14 and 15, a compare function of the IOE is described. A user selects two diagrams: a diagram previously imported into Visio and a diagram to be imported. One will be the “BASE” and the other will be the “CHANGED”. Comparisons can be done from any View Option combination. Embodiments preferably include two different methods for comparing; 1) Melding, where all objects from both diagrams are displayed and color coded to indicate which are Common, which are New and which were on the Base diagram only (FIG. 14); and 2) Twin Screen, where both the diagrams are displayed with the same color code representing the change in Object from Base to Changed (FIG. 15).” Keenan at 19:17-29. “Characteristics from Structural Metadata Repositories for the respective object in the IOE are preferably loaded into the IOE on a daily basis. History is preferably saved for a defined period. The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. The IOE preferably allows queries originating in Structural Metadata repositories to query the topology in the IOE.” Keenan at 21:10-17.
--	---

	 Keenan at Fig. 1. “It is understandable to those skilled in the art that all or a part of steps of the processes in the preceding embodiments is preferably implemented by relevant computing hardware instructed by a program. The program may be stored in a computer readable storage medium. The storage medium may include a ROM, a RAM, a magnetic disk or a compact disk.” 21:27-33.
1[g]. determining using the one or more computing devices, for the subset of third party diagram components, correlated threat model components as defined in the one or more mapping files;	Keenan discloses this claim limitation. For example, Keenan discloses: “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102

	into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function.” 18:23-28. “Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.” Keenan at 18:29-39. “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50. “Turning to FIGS. 14 and 15, a compare function of the IOE is described. A user selects two diagrams: a diagram previously imported into Visio and a diagram to be imported. One will be the “BASE” and the other will be the “CHANGED”. Comparisons can be done from any View Option combination. Embodiments preferably include two different methods for comparing; 1) Melding, where all objects from both diagrams are displayed and color coded to indicate which are Common,
--	---

	which are New and which were on the Base diagram only (FIG. 14); and 2) Twin Screen, where both the diagrams are displayed with the same color code representing the change in Object from Base to Changed (FIG. 15).” Keenan at 19:17-29. “Characteristics from Structural Metadata Repositories for the respective object in the IOE are preferably loaded into the IOE on a daily basis. History is preferably saved for a defined period. The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. The IOE preferably allows queries originating in Structural Metadata repositories to query the topology in the IOE.” Keenan at 21:10-17.  FIGURE 1 Keenan at Fig. 1.
1[h]. displaying on the one or more user interfaces a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and	Keenan discloses this claim limitation. For example, Keenan discloses: “These diagrams depict data flows linking together these information assets, for which the final product is calculated output (“calcs”) used for a specific business purpose. These diagrams are loaded into a repository that stores, reveals and allows for maintenance of the topology. Embodiments include a standard of diagramming for classifying the objects and defining the format of the diagram.” Keenan at 1:24-33 “d) Diagrams are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams are not stored in

	the IOE; instead the diagram serves as a communication medium to and from users and the IOE. e) There is a one to one relationship between an object class and a diagram shape. f) The diagramming standards are pre-specified sets of object shapes and formats and other visual elements. g) Diagrams created in the IOE automatically synchronize with same intelligence stored in structural metadata associated with the object; objects are related to other objects through “links” that obey pre-set rules, thus establishing the logic of the topological structure and enforcing integrity within the system.” Keenan at 2:3-17 “Diagram: Communication vehicle between user and the IOE and between the IOE and the user depicting static topological information in a format selected by, and consumable by users which can be set in documentation, stored as a file in other applications. This term contrasts with the term for the visual depiction of topology seen in the GUI of the IOE which is referred to as a “View”. Another way to understand a diagram is as a fixed portion of the topology. A diagram is topology packaged to be communicated.” Keenan at 4:5-14 “MLD (Model Level Diagram): Type of IOE diagram centered around a specific model. The purpose of this diagram is to depict feeds into the model from all sources (including other models) and the destination of the output (including other models). MLDs are joined together to form Stream Diagrams. Module: internal component of a model which produces output consumed within the model. Path: A sequential set of objects connected by data flow without a fork. It is defined by a single object on the upstream end and a single object on the downstream end. It is possible to have a multiple paths starting at the designated upstream object and ending at the designated downstream object although this would indicate a probable misconception of the diagram or process.” Keenan at 4:46-60 “Topology: Representation of the relationships between objects in an analytical infrastructure. In the case where analytical infrastructure produces information, these relationships represent the storage, processing and flow of information. Communicating topology to and from the IOE is done with diagrams.” Keenan at 5:5-10.
--	---

	“Task Sequence (Topology): Depicts tasks in process order sequence including tasks that require use of information assets.” Keenan at 5:17-19. “Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:40-44. “View: Depiction of topology in the IOE via GUI. A view that is exported or printed from the IOE is a “Diagram”. View Box: that portion of the IOE GUI screen that contains the View.” Keenan at 5:24-27 “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “Platform Objects include physical and/or software systems that house data storage objects, models, or delivery tools, or that support the production of data flow objects.” Keenan at 7:60-62 “topology(x) extracts from the overall topology, a sub-topology based on an object list x, or other subsetting criterion. This function returns a list-object with the appropriate object-level information and formatting to support the generation of visual diagrams and other similar analyses. As with many of the basic functions, the list preferably includes a time/date stamp since the topology is changing in real time.” Keenan at 9:56-63
--	--

	“Embodiments employ direct nesting of internal functions as in the following example: dplot(topology(supports(CCAR.engine))) where dplot() is a function that generates a visual diagram for an infrastructure subset, topology() is, as described above, a function that creates an annotated list of identified objects, where that annotation includes the required information for graphical display and/or other types of analyses, and supports (), as described above identifies all of the objects supporting an end-use reporting tool. Thus, in the example, if ‘CCAR.engine’ is a tool that creates official CCAR reports, then this one-line statement results in the drawing of a diagram for the entire analytic infrastructure that supports CCAR reporting.” Keenan at 10:37-50 “In embodiments of the present system, there are preferably four modes for the user to interact with the IOE via a Graphical User Interface (GUI): 1) a View mode, where the user can view any section of the topology and Characteristics by Object; 2) a Modify mode, where the user can add, change and delete Objects or Characteristics; 3) a Compare mode, where the user can visualize the difference between two diagrams; and 4) a Query mode, where the user can interrogate the topology and Characteristics using standard queries and user-generated queries (script writing). Query results can be tabular (reports) or diagrammatic results in which case the diagram will appear in the View Box. Each mode brings up a different screen format. The portion of the GUI that the user sees is referred to herein as the “screen” as opposed to the logic which dictates what appears on the screen which would be the unseen parts of the GUI. Turning to FIGS. 2 and 3, an exemplary GUI for a View mode is now described, in accordance with an embodiment. The screen contains a view box 202 that occupies most of the screen. Within the view box the user is presented with the topology to view, modify or compare. The view box is view-only in the View and Compare modes, while it is interactive in the Modify mode.” Keenan at 11:42-64 “The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function.” 18:23-28.
--	--

	“Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.” Keenan at 18:29-39. “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50. “To connect objects on the screen, the GUI provides Flow Objects, which preferably connect to any point on the Object shape in the diagram to allow for clarity in the depiction of the topology. Double clicking on the object opens the characteristics table. Double clicking on the Characteristic permits the user to make necessary changes and save. The GUI further preferably permits en masse changes (e.g. property change for entire object class) by selecting an object class, and selecting a class change option.” Keenan at 18:66-19:5 “Turning to FIGS. 14 and 15, a compare function of the IOE is described. A user selects two diagrams: a diagram previously imported into Visio and a diagram to be imported. One will be the “BASE” and the other will be the “CHANGED”. Comparisons can be done from any View Option combination. Embodiments preferably include two different methods for comparing; 1) Melding, where all objects from both diagrams are displayed and color coded to indicate which are Common, which are New and which were on the Base diagram only (FIG. 14); and 2) Twin Screen, where both the diagrams are displayed with the same color code representing the change in Object from Base to Changed (FIG. 15).” Keenan at 19:17-29.
--	---

“Characteristics from Structural Metadata Repositories for the respective object in the IOE are preferably loaded into the IOE on a daily basis. History is preferably saved for a defined period. The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. The IOE preferably allows queries originating in Structural Metadata repositories to query the topology in the IOE.” Keenan at 21:10-17.

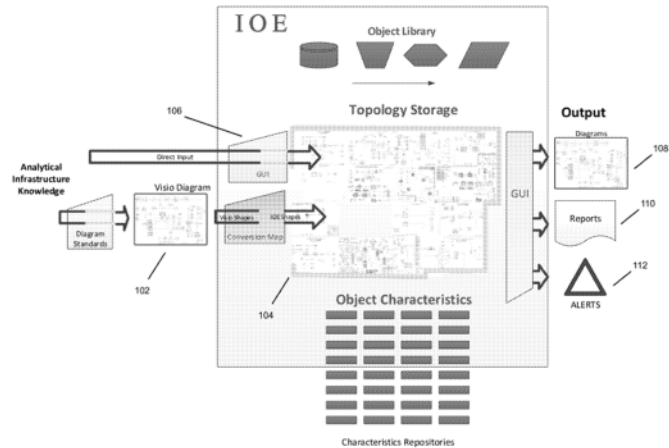


FIGURE 1

Keenan at Fig. 1.

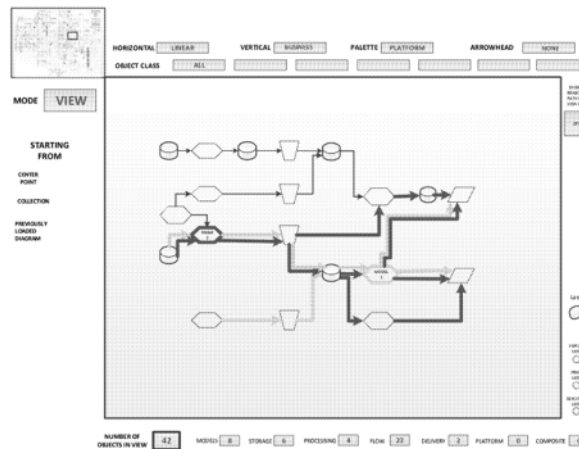


FIGURE 10

Keenan at Fig. 10.

1[i]. generating, using the one or more computing devices, and displaying, on

Keenan discloses this claim limitation. For example, Keenan discloses:

the one or more user interfaces, a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.

“Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:40-44.

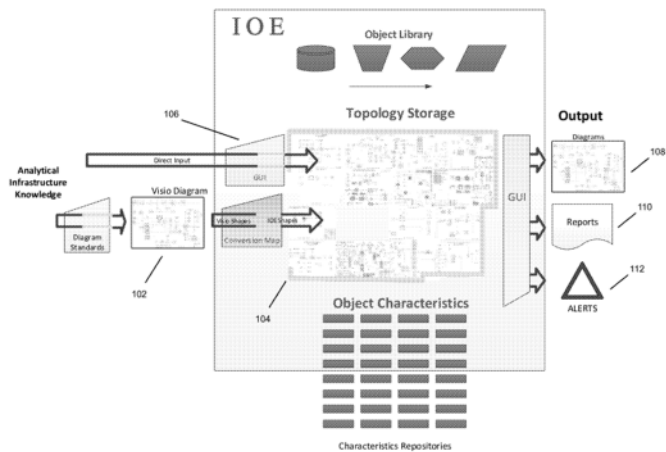


FIGURE 1

Keenan at Fig. 1.

“It should be noted that reports, such as tables, data files, charts or graphs that are generated by reporting systems, which may also store the supporting data and the generated reports, would be considered composite objects, as described below.” Keenan at 8:15-20

“The GUI of the IOE preferably includes a variety of interactive functions for the user. One such interactive function is path highlighting, described herein with reference to FIG. 9. The user clicks on an object in the topology on the screen and the View then highlights the upstream and downstream flows into and out of the object selected. The highlighting is preferably accomplished by using the thicker line weighting on the Flow Objects in different color than normally used for lines. Clicking again on the object toggles this feature off. Paths upstream and downstream from up to six objects can be toggled on at the same time. A different highlighting color may be used for each. The GUI further permits selection of multiple objects for path highlighting, as shown in FIG. 10. The Path Highlight can be converted into a Report by clicking a report icon 902 on the screen. Clicking this icon prompts a box to appear asking the user if they would like the Upstream Paths, Downstream Paths

	or both. Following this selection, the user is asked to select which object classes they wish to include in the report. Following this selection the user is asked which Characteristics of the Object Classes selected would they like on the report. The GUI also preferably includes an option for viewing a full list of the characteristics of an object by double (left) clicking the object. Further, double-clicking on a Characteristic brings up the company's Business Glossary entry for that term. The list appears with an "x" in the upper right corner to allow the user to close the list. The Characteristic list can be converted into a Report by clicking a report icon on the screen." Keenan at 17:11-40. "The GUI also preferably includes an option for viewing a full list of the characteristics of an object by double (left) clicking the object. Further, double-clicking on a Characteristic brings up the company's Business Glossary entry for that term. The list appears with an "x" in the upper right corner to allow the user to close the list. The Characteristic list can be converted into a Report by clicking a report icon on the screen." Keenan at 17:41-52. "Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View; Depict Riskiest Path, which depicts the Path whose objects in aggregate measure riskiest versus all other Paths in the View, using a formula preferably based on Risk value characteristics of the Objects; Depict Lowest Data Quality Path, which depicts the Path whose data elements in aggregate measure lowest in data quality versus all other Paths in the View, using a formula preferably based on Data Quality value characteristics of the Data Elements; Calculate Aggregate risk by View, which depicts risk values of the objects of the View aggregated based on an aggregation formula." Keenan at 18:6-22. "Embodiments of the present system include advanced reporting capabilities. In addition to standard reports discussed above, special reports can be created using the GUI. Examples of such reports include Tree Pair File Reports and Lasso Reports. A Tree Pair File Report identifies pairs of objects of the same class upstream of the selected object. These pairs are referred to as
--	---

“Tree Pairs.” A Tree Pair is defined as a pair of objects, connected via a Data Flow object through any combinations of objects. The user selects the one or more objects for the query to determine the Tree Pairs. These selected objects will be displayed in the first column of the report. Working upstream from the selected object, each object of the same class, regardless of any object(s) between the selected model and itself, is considered a part of the tree of the selected object's class. In succession, the query continues looking upstream from each object in the same object class in the Tree for the next object. This in turn creates an object Pair. The query continues upstream along each Path and Branch of the tree until the last object of the same class is found. The collection of Pairs for a the selected model are referred to as Tree Pairs of object class of “x” where “x” is a given selected model. The collection of objects that feed the selected object may be connected linearly or in a forked path. If there is a loop (a sequence in which an object appears more than once), then the sequence is stopped at the second instance of that object.” Keenan at 19:30-65.

“The GUI also permits the creation of Lasso Reports, which are reports which use as the select criteria the object set bounded inside an irregular shape drawn by a “freeform” tool (same as the Visio tool called “freeform”). This topology subset can be converted into a Report by clicking a report icon on the screen. Clicking this icon, the user will be asked to select which object classes they wish to include in the report. Following this selection, the user is asked which Characteristics of the Object Classes selected would they like on the report.” Keenan at 19:56-65.

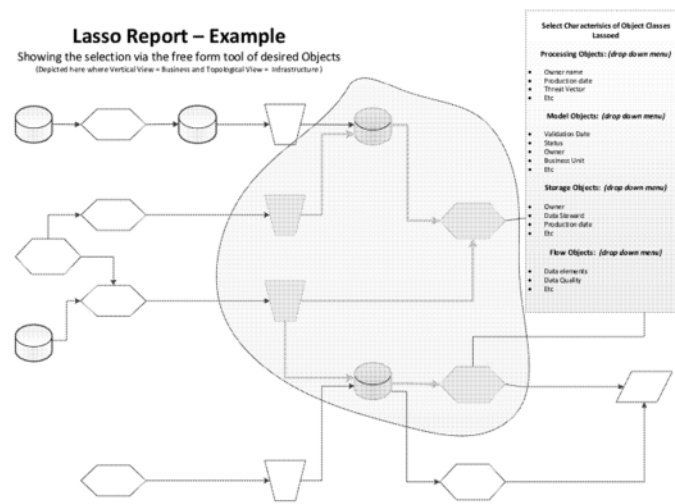


FIGURE 16

	Keenan at Fig. 16.
2. The method of claim 1, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.	Keenan discloses this claim limitation. For example, Keenan discloses: “It is understandable to those skilled in the art that all or a part of steps of the processes in the preceding embodiments is preferably implemented by relevant computing hardware instructed by a program. The program may be stored in a computer readable storage medium. The storage medium may include a ROM, a RAM, a magnetic disk or a compact disk.” Keenan at 21:27-33.
3. The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.	Keenan discloses this claim limitation. For example, Keenan discloses: “To connect objects on the screen , the GUI provides Flow Objects , which preferably connect to any point on the Object shape in the diagram to allow for clarity in the depiction of the topology . Double clicking on the object opens the characteristics table.” 18:65-19:2. “It should be noted that reports , such as tables , data files , charts or graphs that are generated by reporting systems , which may also store the supporting data and the generated reports , would be considered composite objects, as described below.” 8:16-20 “The GUI also provides the user with an option to have the output appear in a table which can be exported as excel file.” 13:53-55.
4. The method of claim 1, wherein the data file comprises one of a VSD file, a VSDX file, a TMT file, an XML file, a JSON file, a PNG file, and a JPEG file.	Keenan discloses this claim limitation. For example, Keenan discloses: “The objects and characteristics are stored in a graphical format 102 such as a Visio file.” Keenan at 5:36-37. See also https://en.wikipedia.org/wiki/Microsoft_Visio
5. The method of claim 1, wherein determining, for the subset of third party diagram components, correlated threat model components, comprises first determining whether a user-defined name for one	Keenan discloses, or at least renders obvious, this claim limitation. For example, Keenan discloses: “Each object preferably has two identifiers: one internal to the IOE, and one that ties the object to an authoritative external repository. For example, all models in a company inventory (current, historical, and to some extent prospective) may be entries in one external data model characteristics repository

of the subset of third party diagram components has a match in the one or more mapping files and, if no match is found, then determining whether a default title for the third party diagram component, as defined by the third party software application, has a match in the one or more mapping files.	system, along with a wide variety of descriptive data and metadata. Therefore, all “model” objects in the IOE preferably carry their repository IDs as well as an ID internal to the IOE. By extension, all objects in the IOE, with the exception of data flow objects are associated with an authoritative external information repository, and linked directly to that repository by an ID unique to that repository. Objects without such a linkage are under-identified. The external identifier is preferably flexible to accommodate any potential diversity.” Keenan at 6:13-26 “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future. To add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found.” Keenan at 18:36-60. “To connect objects on the screen, the GUI provides Flow Objects, which preferably connect to any point on the Object shape in the diagram to allow for clarity in the depiction of the topology. Double clicking on the object opens the characteristics table. Double clicking on the Characteristic permits the user to make necessary changes and save. The GUI further preferably permits en masse changes (e.g. property change for entire object class) by selecting an object class, and selecting a class change option.” Keenan at 18:65-68; 19:1-6.
--	---

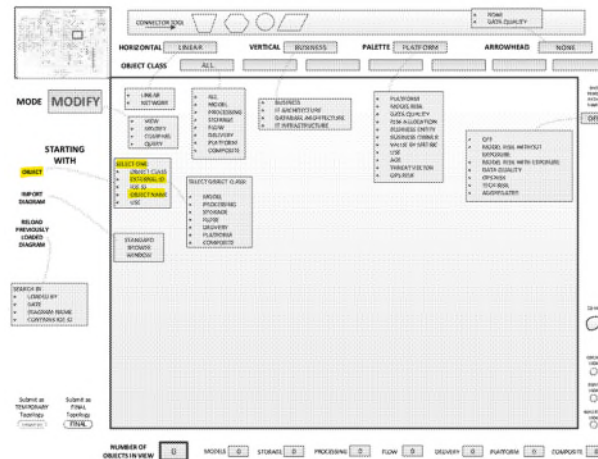


FIGURE 13

Keenan at Fig. 13

6. The method of claim 5, further comprising, if no match is found for the default title, matching the third party diagram component with a placeholder component and displaying the placeholder component on the relational diagram to indicate that no match was found.

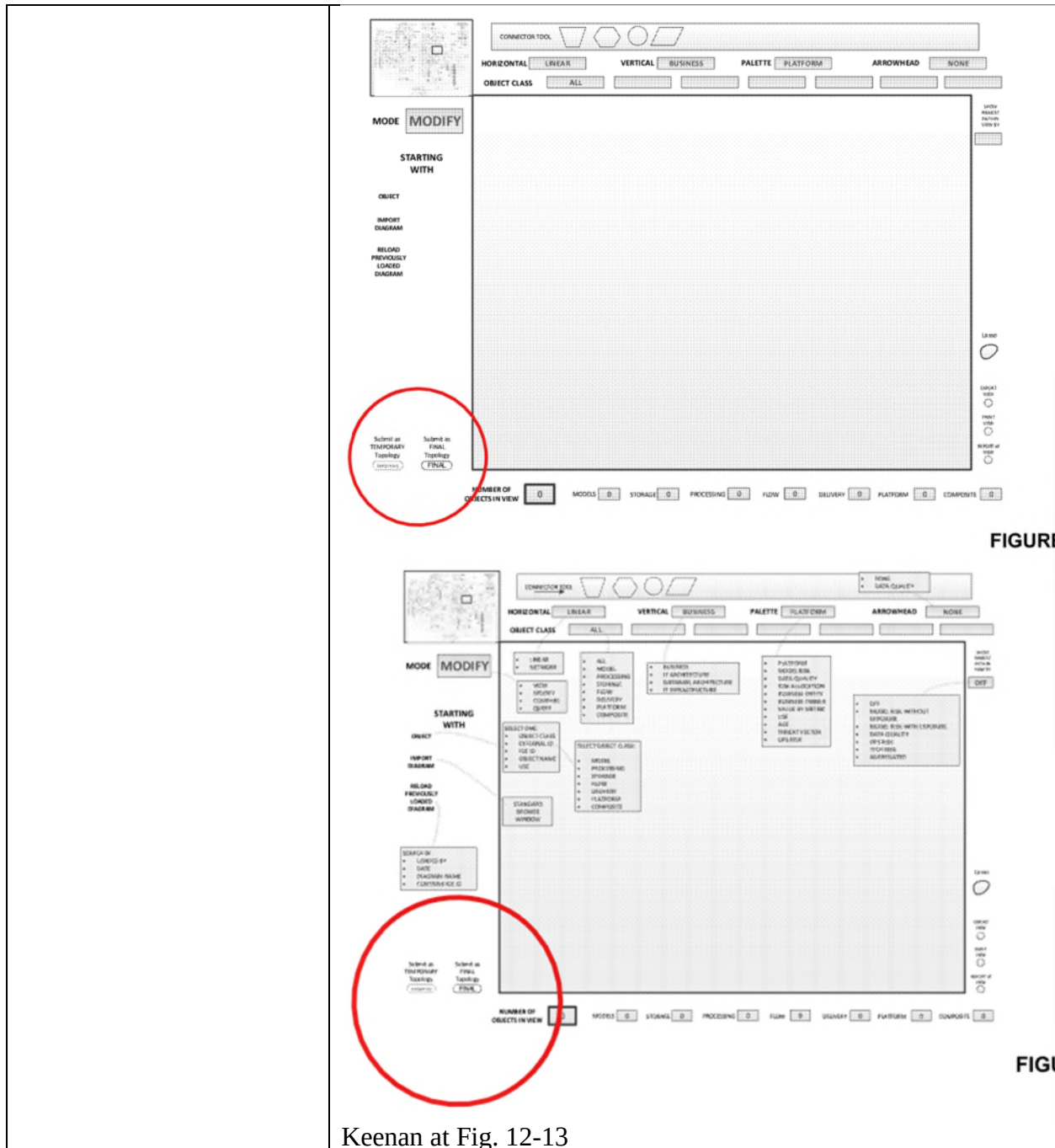
Keenan discloses, or at least renders obvious, this claim limitation. For example, Keenan discloses:

“WHERE MODELID IN list preferably returns a list of model owners corresponding only to models nested in the platform whose ID is 33, and leaving list as an object in temporary memory. Alternatively, and avoiding the preservation of an intermediate object in temporary memory, the condensed statement,” Keenan at 11:7-11

“Instead, the temporary topology, in whatever manner it was modified, is saved in a Temporary topology and the user modifying can name the temporary saved topology, such that any changes between existing and temporary modifications are highlighted. Further, in order to preserve past states of the topology, the IOE preferably stores the history of the full IOE topology on a periodic basis.” Keenan at 19:9-16

“The GUI also provides the user with an option to have the output appear in a table which can be exported as excel file. If the criteria entered is not currently stored in the IOE, the user is given the option to add new topology, described more fully below. With a Single Object Selection, the user may specify a Use, ObjectID and/or Object Name to identify an object, along with all objects feeding into and all objects being fed by the Object selected. With a Multiple Object Selection, the GUI permits the user to specify a Business entity, a diagram previously imported, a person's name, and/or an object class or subclass. Once the selection is made, the GUI permits the user to

	refine the view by filtering on a variety of criteria.” Keenan at 13:53-65 “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50. “To add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found.” Keenan at 18:51-60.
--	--



Keenan at Fig. 12-13

7. The method of claim 1, further comprising determining, for the one or more relationships defined by the data file, correlated threat model components as defined in the one or more mapping files.

Keenan discloses this claim limitation. For example, Keenan discloses:

“In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a

	standard, canonical format for storage in a topology graph database 104 of the infrastructure.” Keenan at 5:33-40. “According to embodiments of the present invention, an Integrated Object Environment (IOE) running in a graph database environment is used. Diagrams of the production chain of information assets are created. These diagrams depict data flows linking together these information assets, for which the final product is calculated output (“calcs”) used for a specific business purpose. These diagrams are loaded into a repository that stores, reveals and allows for maintenance of the topology.” Keenan at 1:24-30 “Junction: Multiple Paths join in at a single object and multiple Paths exit from this object. Junction Problem: Situation where multiple Paths join at a Junction but no information from one or more upstream Paths is conveyed, in any form, by the downstream Paths. This creates a problem for tracking the data lineage and requires that data elements be characteristics of Data Flow objects in order to trace the flow of information through a Junction.” Keenan at 4:39-45 “Module: internal component of a model which produces output consumed within the model. Path: A sequential set of objects connected by data flow without a fork. It is defined by a single object on the upstream end and a single object on the downstream end. It is possible to have a multiple paths starting at the designated upstream object and ending at the designated downstream object although this would indicate a probable misconception of the diagram or process.” Keenan at 4:54-60 “The data flow between the objects is also defined as an object class. The object classes and links have features that allow for its characterization including specification of its state. The relationships characterizing the links between objects are preferably fixed (non-modifiable), to accommodate a restriction on data flow objects feeding into other data flow objects. This environment permits storage of the objects that represent the analytic infrastructure, and the structural metadata. A scripting language is preferably adopted and an interface designed that allows users to build and store/run complex programming statements that apply logic directly to the topology or any subset thereof.” Keenan at 5:51-62
--	--

	“While data flow objects have direction, with the source and destination information represented in the topology, direction is implicitly included and so an explicit direction feature does not need to be embedded in the objects themselves. Data element level information preferably identifies whether it is “datum” or is “calc”. Data flow objects link to downstream objects using a “passes to” concept in the data flow object, as opposed to characterizing the linkage as “receives from” in the downstream object.” Keenan at 7:25-34 “Composite Objects are composites of multiple objects designed to accommodate multiple class characteristics. For example, a spreadsheet may behave as both a data storage object and as a model. Essentially, the composite object acts as a wrapper, providing the necessary information to keep the underlying objects fully and independently identified, for each object class it is a member of, and providing an association between class characteristics and the components of the composite object. That is, to re-use the multi-function spreadsheet example above, the composite object would be a dual list of defining characteristics for ‘models’ and ‘data source’ objects, and provide a method by which data flow objects could be separately linked to the components. More specifically, some data may flow into the object only to be stored and later fed into some downstream objects and that data flow object needs to exist independently and be fully identified. At the same time, another data flow object may flow into the composite purely to feed the model never to be re-directed to any other objects, and that data flow object needs to exist independently and be fully identified.” Keenan at 8:21-41 “Turning to FIG. 4, a Horizontal View is described, in accordance with an embodiment. The GUI permits the Horizontal View to be shown in multiple ways. For example, a Linear View 402 depicts the production chain of information assets as the data flow sequentially progresses downstream through the chain of information assets. This Linear View is created by following the path of data elements in the data flow objects as they are passed from object to object. Since objects are stored in the IOE as a single instance, the multiple loads and extracts to the same database in a View are depicted as unclassified objects in the topological space allowing depiction of the proper sequence of the production chain. These Unclassified Objects are created dynamically whenever a Linear View is requested. The Characteristics of these dynamically created Unclassified Objects equals the Characteristics of the
--	--

	Storage Object for which it represents an instance of. For example, if model X produces calc “A” and calc “A” is subsequently loaded to database STS, and model Y extracts calc “A” from STS, then by inference, model X and model Y are on the same production chain and X is upstream of Y. On the Linear View, Model X will be connected to Model Y with a Data Flow object to a Unclassified Object representing the storage in STS followed by a Data Flow object connecting the Unclassified Object to Model Y. If STS is loaded and extracted from multiple locations along the production chain, then a proxy shape will be depicted on the View at each load and extract point to maintain production chain continuity.” Keenan at 15:15-43. “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50.
8[p]. A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:	“It is understandable to those skilled in the art that all or a part of steps of the processes in the preceding embodiments is preferably implemented by relevant computing hardware instructed by a program. The program may be stored in a computer readable storage medium. The storage medium may include a ROM, a RAM, a magnetic disk or a compact disk.” Keenan at 21:26-33 See Elements 1[p]-1[a] and 1[e].
8[a]. a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the	See Element 1[b].

threat model components through the one or more databases;	
8[b]. one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);	See Elements 1[c] and 1[d].
8[c]. an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by one of the one or more computing devices, the data file generated by the third party software application, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;	Keenan discloses this claim limitation. For example, Keenan discloses: See Elements 1[e] and 1[f]. “A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.” Keenan at Abstract. “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as

	diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44. “Within a screen of the GUI of the IOE is preferably a box which allows the user a quick way to calculate the risk of the selected View. Clicking this box creates both a visual depiction of the information requested as well as giving the user the option of printing the report. Examples of those risk reports include: Aggregate Model Risk of the View, which calculates the aggregate model risk for all models in the View; Depict Riskiest Path, which depicts the Path whose objects in aggregate measure riskiest versus all other Paths in the View, using a formula preferably based on Risk value characteristics of the Objects; Depict Lowest Data Quality Path, which depicts the Path whose data elements in aggregate measure lowest in data quality versus all other Paths in the View, using a formula preferably based on Data Quality value characteristics of the Data Elements; Calculate Aggregate risk by View, which depicts risk values of the objects of the View aggregated based on an aggregation formula. The GUI of the IOE also preferably includes an option to export. Any View created in the IOE is exportable to a graphical design program, such as Microsoft Visio, by clicking on the Export button on the right side of the View Box. The map of Visio shapes to IOE shapes is the same as the import function.” 18:6-35. “The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise's analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 1. In general, each real world component in the infrastructure is defined as an object with associated characteristics such as state, and its relationship to other objects. The objects and characteristics are stored in a graphical format 102 such as a Visio file. The system converts the Visio files 102 into a standard, canonical format for storage in a topology graph database 104 of the infrastructure. Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:27-44
--	---

	“Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.” Keenan at 18:29-35. “A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, ‘The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.’ The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future.” Keenan at 18:36-50. “Turning to FIGS. 14 and 15, a compare function of the IOE is described. A user selects two diagrams: a diagram previously imported into Visio and a diagram to be imported. One will be the “BASE” and the other will be the “CHANGED”. Comparisons can be done from any View Option combination. Embodiments preferably include two different methods for comparing; 1) Melding, where all objects from both diagrams are displayed and color coded to indicate which are Common, which are New and which were on the Base diagram only (FIG. 14); and 2) Twin Screen, where both the diagrams are displayed with the same color code representing the change in Object from Base to Changed (FIG. 15).” Keenan at 19:17-29. “Characteristics from Structural Metadata Repositories for the respective object in the IOE are preferably loaded into the IOE on a daily basis. History is preferably saved for a defined period. The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. The IOE preferably allows queries originating in Structural Metadata repositories to query the topology in the IOE.” Keenan at 21:10-17.
--	--

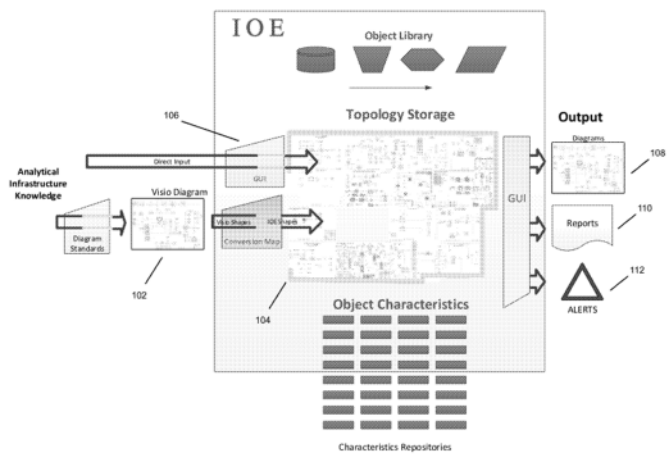


FIGURE 1

Keenan at Fig. 1.

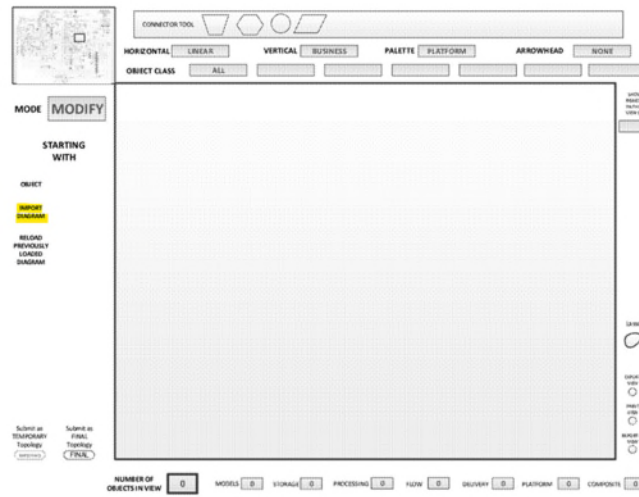


FIGURE 12

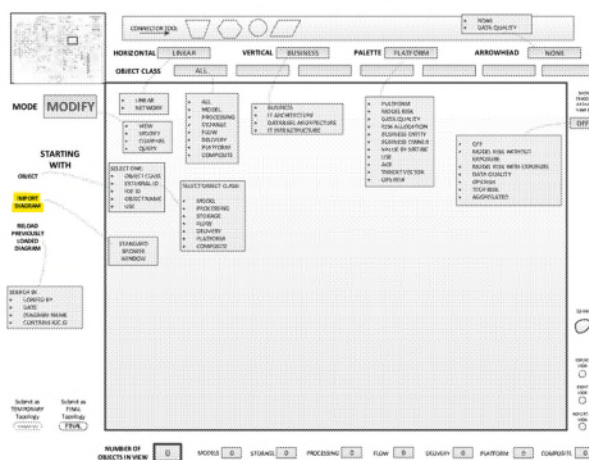
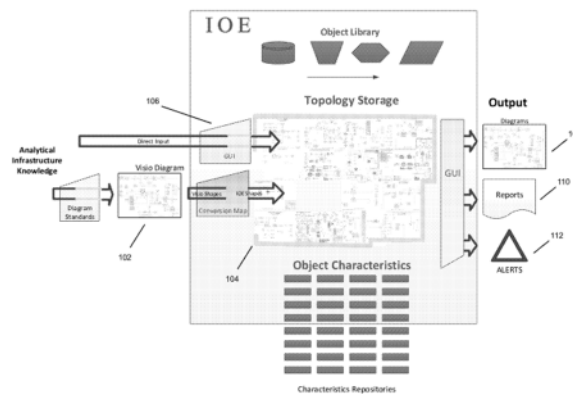
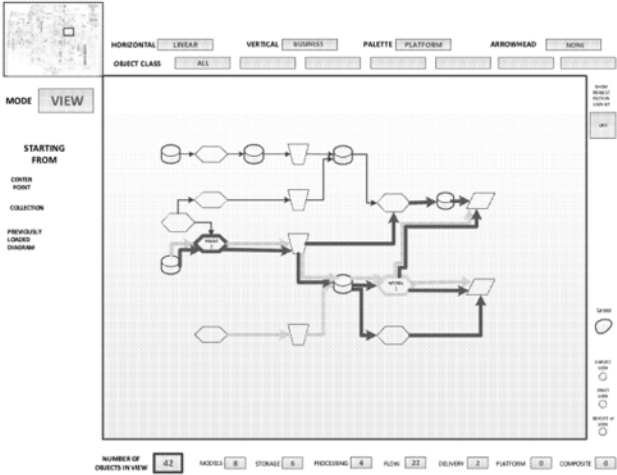


FIGURE 13

	Keenan at Fig. 12-13.
8[d]. an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;	See Element 1[g]. “In embodiments of the present system, there are preferably four modes for the user to interact with the IOE via a Graphical User Interface (GUI):...” Keenan at 11:42-43 see Elements 1[e], 8[c].
8[e] a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of one of a system, an application, and a process, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and	See Element 1[h]. “d) Diagrams are rendered by a front end/GUI of the IOE and are both input and output of the IOE. Diagrams are not stored in the IOE; instead the diagram serves as a communication medium to and from users and the IOE.” Keenan at 2:3-6 “These available options are more fully illustrated in FIG. 3, which shows options available through drop-down menus of user-selectable items on the screen in the IOE GUI's View mode, in accordance with an embodiment.” Keenan at 15:11-14  FIGURE 1

	 FIGURE 10
8[f]. a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.	Keenan at Fig. 1 and 10 See Element 1[i]. “Alternatively, a GUI 106 is provided for inputting directly into the topology database. Rendering software (GUI) 106 converts the graph database and associated characteristics into Views, which are exportable as diagrams 108, as well as reports 110 and alerts 112.” Keenan at 5:40-44
9. The system of claim 8, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.	See Claim 2.
10. The system of claim 8, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.	See Claim 3.
11. The system of claim 8, wherein the data file comprises one of a VSD file, a VSDX file, a TMT File, an XML file, a JSON	See Claim 4.

file, a PNG file, and a JPEG file.	
12. The system of claim 8, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.	Keenan discloses, or at least renders obvious, this claim limitation. For example, Keenan discloses: “Each object preferably has two identifiers: one internal to the IOE, and one that ties the object to an authoritative external repository. For example, all models in a company inventory (current, historical, and to some extent prospective) may be entries in one external data model characteristics repository system, along with a wide variety of descriptive data and metadata. Therefore, all “model” objects in the IOE preferably carry their repository IDs as well as an ID internal to the IOE. By extension, all objects in the IOE, with the exception of data flow objects are associated with an authoritative external information repository, and linked directly to that repository by an ID unique to that repository. Objects without such a linkage are under-identified. The external identifier is preferably flexible to accommodate any potential diversity.” Keenan at 6:12-26. “To add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found.” Keenan at 18:51-60.
13. The system of claim 8, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.	Keenan discloses, or at least renders obvious, this claim limitation. For example, Keenan discloses: “Each object preferably has two identifiers: one internal to the IOE, and one that ties the object to an authoritative external repository. For example, all models in a company inventory (current, historical, and to some extent prospective) may be entries in one external data model characteristics repository system, along with a wide variety of descriptive data and metadata. Therefore, all “model” objects in the IOE preferably carry their repository IDs as well as an ID internal to the IOE. By extension, all objects in the IOE, with the exception of data flow objects are associated with an authoritative external information repository, and linked directly to that repository by an ID unique to that repository. Objects without such a linkage are under-identified. The external identifier is preferably flexible to accommodate any potential diversity.” Keenan at 6:12-26.

	“In an embodiment, a straightforward object identification and filtering syntax is provided. For consistency with common object oriented programming, the closest set of square brackets is used to indicate identity as follows: datastorageobject[33, ioeid=F], or equivalently, datastorageobject[33] returns the data storage object whose external ID is 33. Here ioeid holds the default value F. Hence datastorageobject[100, ioeid=T] returns the data storage object whose internal ID is 100. Because the objects are themselves lists, a component extraction syntax is preferably used. Again, for consistency with common object oriented programming, a the dollar sign \$ is preferably to indicate an object component. For example, if ‘model.name’ is an included characteristic of model objects, and the model with an external ID of 23456543 is named “European Windstorm Model”, then model[23456543]\$model.name returns “European Windstorm Model”. Similarly, if ‘location’ and ‘owner’ are both characteristics of a ‘server’ sub-class of platform, then the statement, output<-platform[platform.type==“server”, c(\$ioeid, \$location, \$owner)] returns a list of all servers identified by ID, location, and owner.” Keenan at 9:29-53. “The GUI also preferably includes an option for searching diagrams previously loaded into the IOE database. The results of the search are preferably returned in tabular form, and include the External IDs for the list of diagrams where the model was loaded, thus permitting a user to select a diagram to view. Alternatively, the search results are returned by Business Affiliation (e.g Business Unit), or by the Name of Object (e.g., the user first selects the type of object they are looking for, then enters the name). The IOE then return closest results if no exact match found. The results preferably include the Name of Object, the object's load date, and the name of the person who imported the object into the IOE database. The GUI of the IOE also preferably includes links to external documents. In this manner, an icon is presented within an object to indicate that a document electronically stored outside of the IOE has been linked to the object. The icon represents the document format. By clicking on this icon, the application which supports the document opens and the document appears on the screen.” Keenan at 17:53-17:67;18:1-5.
--	--

	“A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, “The following objects matched existing objects from the IOE, new relationships will be loaded to existing object.” The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future..” Keenan at 18:36-50. “To add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found.” Keenan at 18:51-60
14. The system of claim 8, wherein, for each of the subset of third party diagram components for which no correlated threat model component is found in the one or more mapping files, the diagram interface displays a placeholder component indicative of a match not being found.	See Claim 6.
15. The system of claim 8, further comprising one or more mapping interfaces displayed on the one or more computing devices and configured to, in response to receiving user	Keenan discloses, or at least renders obvious, this claim limitation. For example, Keenan discloses: See Element 1[e]. “A system and method are described to create and use an Integrated Object Environment (IOE) running in a graph

input, modify the one or more mapping files.	database environment. Uses include storing, revealing and maintaining value and risk of information assets, such as the topology of an analytical infrastructure in query-driven, graph database. A graphical user interface is described to permit importation, viewing, modification and querying in the IOE.” Keenan at Abstract “Canvas: Physical screen space in the IOE Modification layer where user builds or modifies topology” Keenan at 4:1-2 “2) a Modify mode, where the user can add, change and delete Objects or Characteristics;” Keenan at 11:46-47 “Within the view box the user is presented with the topology to view, modify or compare. The view box is view-only in the View and Compare modes, while it is interactive in the Modify mode.” Keenan at 11:61-64 “Turning to FIGS. 12 and 13, a modification mode for the IOE is described, in accordance with an embodiment. The GUI for the modification mode is similar to that of the view mode, but permits a user to add objects and connectors using a similar interface and drop-down menus, and through an import function for graphical object files, such as Visio diagrams.” Keenan at 18:29-35 “In embodiments of the IOE, unless saved as FINAL, a modified topology will not be saved in the production topology when the session ends. Instead, the temporary topology, in whatever manner it was modified, is saved in a Temporary topology and the user modifying can name the temporary saved topology, such that any changes between existing and temporary modifications are highlighted. Further, in order to preserve past states of the topology, the IOE preferably stores the history of the full IOE topology on a periodic basis.” Keenan at 19:7-16 “The IOE system administrator can set Acceptance requirements for modifications to the topology. Setting Acceptance Criteria requires the IOE to send an email or other such electronic message to the designated responsible party within a set amount of time of Criteria being met. The designated party then has a set amount of time to Accept the modification. Rejecting, or failing to Accept the modification requires removal of the modification and a notice sent to the modifying party. Both these set times can be set by the IOE system administrator. Examples of Acceptance Criteria include: Model Owners must accept the modification of
---	---

	use of their Model's output; Storage Owners must accept the modification of use of their Storage output; Process Owners must accept the modification of use of their Process output; Any Object owner must accept the addition of or subtraction of Flow Objects to the Object they own." Keenan at 20:61-21:9 "A map is preset that maps Visio shapes and formats to IOE shapes and formats. The user selects the import function in the IOE GUI, which first checks to see if object already exists in IOE based on external ID number first, then name match. If the object already exists, a message is posted to the screen, such as, "The following objects matched existing objects from the IOE, new relationships will be loaded to existing object." The existing Objects are then updated with new relationships based on the imported Visio diagram. The import function also preferably provides utilities for computing and displaying object count by class and total count once diagram is imported, and for saving the diagram objects so that it can be reproduced for audit purposes, so it can be used in diagram comparisons, and to allow users to select as a View in the future. To add an objection directly to the IOE, the modification mode of the GUI includes a screen for the user to select a method for adding an object, such as External Object ID, Use, Business Affiliation, Name of owner, and Name of object. If the IOE returns a topology with the object(s) already stored that match the criteria the user entered, then the user can browse the list to see if the object they want to load is already in the IOE. If not, the user can then add the object. Alternatively, the user can modify the object(s) found." Keenan at 18:36-60.
16[p]. A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more databases, the one or more databases comprising:	See Elements 1[p]-1[a], 1[e], and 8[p].
16[a]. a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components	See Element 1[a] and 8[a].

through the one or more databases;	
16[b]. one or more mapping files communicatively coupled with the one or more databases, the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter “third party diagram components”);	See Elements 1[c], 1[d] and 8[b].
16[c]. an import interface displayed on one of the one or more computing devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by the one or more computing devices, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;	See Element 8[c].
16[d]. an interface configured to, in response to receiving user input, initiate a determination of which threat model components are correlated with the subset of third party diagram components in the one or more mapping files;	See Element 1[g] and 8[d].

16[e]. a diagram interface, displayed on the one or more computing devices, displaying a relational diagram of a computing network, using visual representations of the threat model components correlated with the subset of third party diagram components, the relational diagram defining a threat model; and	See Element 1[h] and 8[e].
16[f]. a threat report interface, displayed on the one or more computing devices, including a threat report displaying each threat that is associated through the one or more databases with one of the threat model components included in the threat model.	See Element 1[i] and 8[f].
17. The system of claim 16, wherein the one or more mapping files are not stored in the one or more databases but are stored on the one or more computing devices.	See Claim 2.
18. The system of claim 16, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.	See Claim 3.
19. The system of claim 16, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.	See Claim 12.

20. The system of claim 16, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.	See Claim 13.
--	---------------