

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”) and Keenan’s Provisional Application US 62/487,370 Filed April 19, 2017 (“Keenan Prov”)
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; Keenan Prov at 1. “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; Keenan Prov at 1. “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; Keenan Prov at 1. “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; Keenan Prov at 2. “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; Keenan Prov at 34. “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; Keenan Prov at 34. “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 Prov at 26-27. “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; Keenan Prov at 5-6. ‘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; Keenan Prov at 15. "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.; Keenan Prov at 16-17.
--	--

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 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

View Name	Options	Description
ARROW-HEAD	None	Arrowheads depicted in solid black color
	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; Keenan Prov at Table 4.1.

“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; Keenan Prov at 19-20.

“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; Keenan Prov at 19-20.

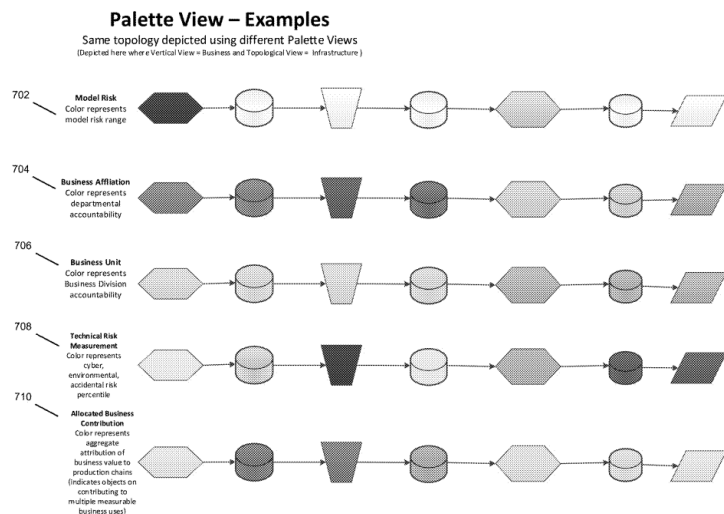


FIGURE 7

Keenan at Fig. 7; Keenan Prov at Fig. 4.4.

	“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; Keenan Prov at 20-21. “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; Keenan Prov at 23. “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; Keenan Prov at 25. “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; Keenan Prov at 31. Lasso Report – Example Showing the selection via the free form tool of desired Objects (Depicted here where Vertical View = Business and Topological View = Infrastructure) Select Characteristics of Object Classes Lassoed Processing Objects: (drop down menu) - Object name - Production date - Model Objects: (drop down menu) - Validation Date - Status - Owner - Business Unit - Etc. Storage Objects: (drop down menu) - Data element - Data element - Data Quality - Etc. Flow Objects: (drop down menu) FIGURE 16 Keenan at Fig. 16; Keenan Prov at Fig. 6.3.
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; Keenan Prov at 1. “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; Keenan Prov at 1.

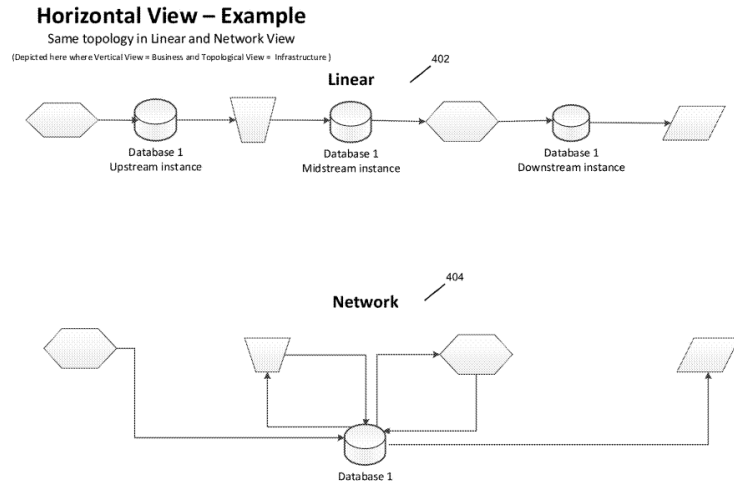


FIGURE 4

Keenan at Fig. 4; Keenan Prov at Fig. 4.1.

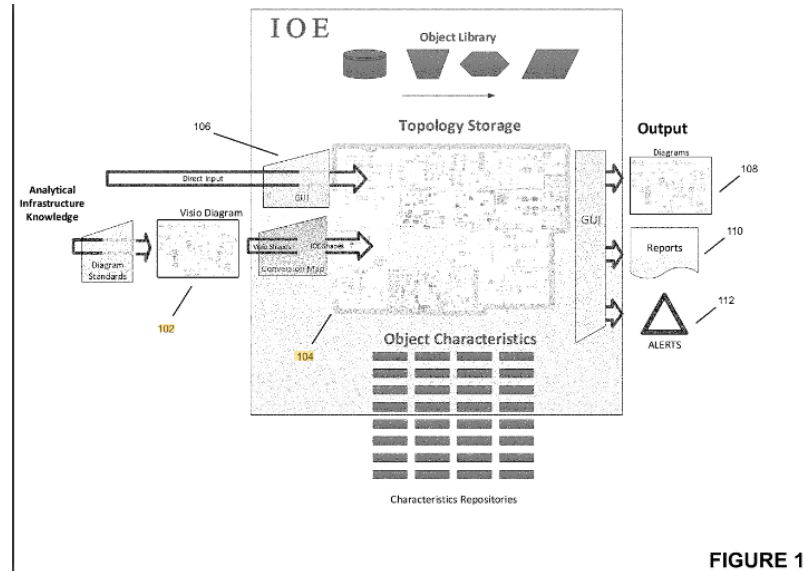
“The IOE, as effected in embodiments, effectively represents a dynamic map of an enterprise’s analytic infrastructure, preferably including all physical assets that contribute to analysis, analytic reporting, and the information created by processing data. The major components of the IOE appear in FIG. 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; Keenan Prov at 26-27.

“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; Keenan Prov at 3. FIGURE 1 Keenan at Fig. 1; Keenan Prov at Fig. 2.1.
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;	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; Keenan Prov at 1. “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; Keenan Prov at 1 “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; Keenan Prov at 1. “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; Keenan Prov at 34. “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; Keenan Prov at 34. “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 Prov at 26-27.



Keenan at Fig. 1; Keenan Prov at Fig. 2.1.

“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 be an inherent characteristic of the object class.” Keenan at 7:41-59; Keenan Prov at 5-6.

‘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; Keenan Prov at 15. "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; Keenan Prov at 17 (Table 4.1). "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; Keenan Prov at 17 (Table 4.1). "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; Keenan Prov at 19-20. "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; Keenan Prov at 19-20.

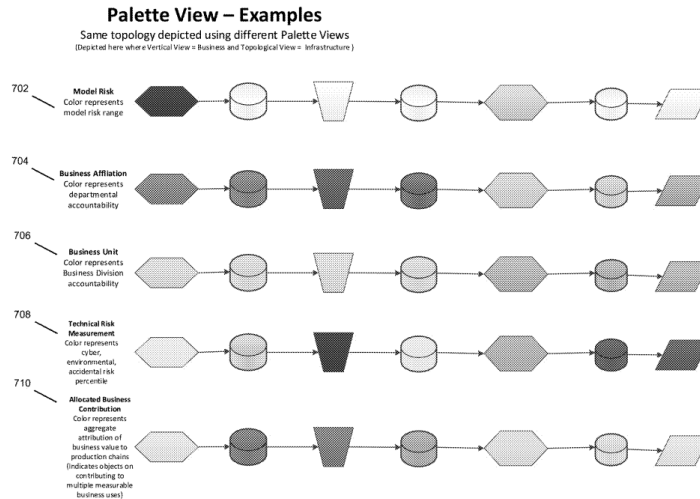


FIGURE 7

Keenan at Fig. 7; Keenan Prov at Fig. 4.4.

“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; Keenan Prov at 20-21.

“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; Keenan Prov at 23. “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; Keenan Prov at 25. “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; Keenan Prov at 31.
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; Keenan Prov at 26-27. “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; Keenan Prov at 13. “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; Keenan Prov at 14.
--	--

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; Keenan Prov at 14.

“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; Keenan Prov at 19-20.

“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; Keenan Prov at 25. “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; Keenan Prov at 26-27. Palette View – Examples Same topology depicted using different Palette Views (Depicted here where Vertical View = Business and Topological View = Infrastructure) FIGURE 7 Keenan at Fig. 7; Keenan Prov at Fig. 4.4.
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; Keenan Prov at 26-27. “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; Keenan Prov at 25.
--	--

	“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; Keenan Prov at 28. “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 Prov at 32. FIGURE 1 Keenan at Fig. 1; Keenan Prov at 2.1.
1[e]. in response to receiving one or more user inputs, using one or more user interfaces displayed	Keenan discloses this claim limitation. For example, Keenan discloses:

on one or more computing devices communicatively coupled with the one or more databases:	“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; Keenan Prov at 1. “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; Keenan Prov at 2. “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; Keenan Prov at 26-27. “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; Keenan Prov at 26. “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; Keenan Prov at 33. “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; Keenan Prov at 35. “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; Keenan Prov at 3. “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; Keenan Prov at 25. “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; Keenan Prov at 12-13. “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; Keenan Prov at 13.
--	--

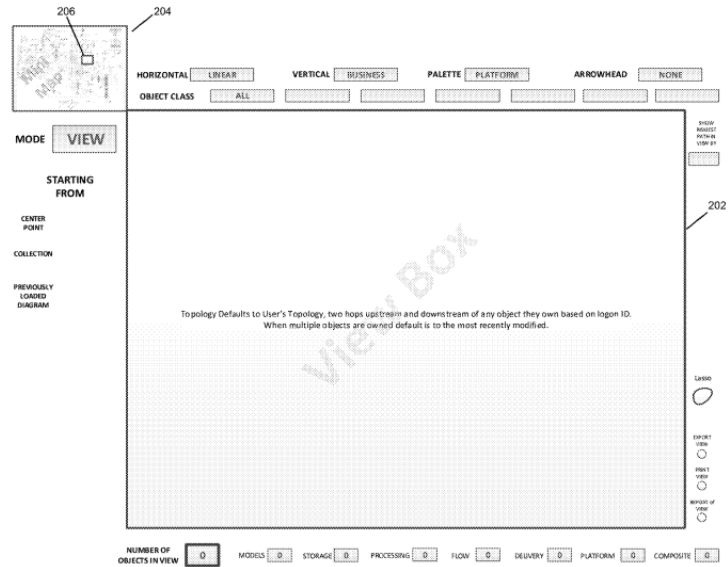


FIGURE 2

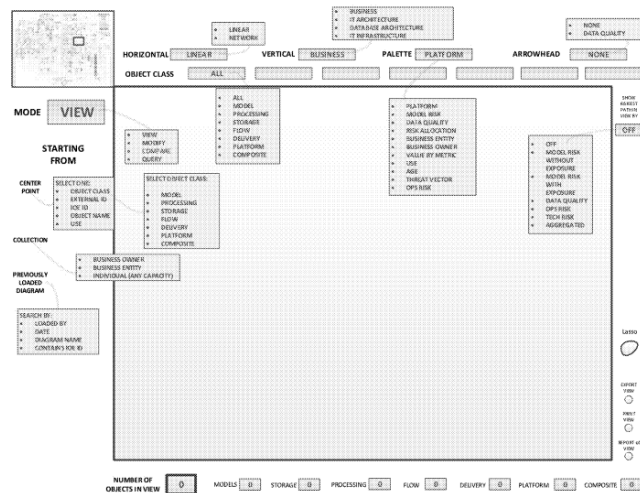


FIGURE 3

Keenan at Figs. 2, 3; Keenan Prov at Figs. 3.1, 3.2.

“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

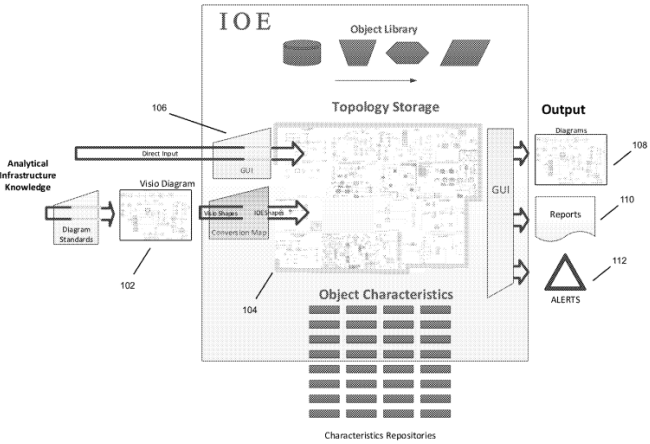
	more fully below. With a Single Object Selection, the user may specify a Use, ObjectID and/or Object Name to identify an object, along with all objects feeding into and all objects being fed by the Object selected. With a Multiple Object Selection, the GUI permits the user to specify a Business entity, 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; Keenan Prov at 16-17. "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 Prov at 3, 17 (Table 4.1), Fig. 4.3. Keenan dedicates a substantial portion <i>See also</i> Keenan at 12:64-14:10; 18:36-50 and Figs. 4-16; Keenan Prov at 26-27.
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; Keenan Prov at 1. "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; Keenan Prov at 34. “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; Keenan Prov at 34. “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; Keenan Prov at 26-27. “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; Keenan Prov at 3. “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; Keenan Prov at 5. “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; Keenan Prov at 6-7. “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; Keenan Prov at 18. “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 Prov at 26-27. “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; Keenan Prov at 25. “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; Keenan Prov at 31-32. “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; Keenan Prov at 26-27. “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; Keenan Prov at 28. “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 Prov at 32.
--	---

	 FIGURE 1 Keenan at Fig. 1; Keenan Prov at 2.1. “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; Keenan Prov at 3, 17 (Table 4.1), Fig. 4.3.
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; Keenan Prov at 26-27.

	“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; Keenan Prov at 25. “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; Keenan Prov at 31-32. “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; Keenan Prov at 26-27. “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; Keenan Prov at 28.
--	--

	“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 Prov at 32. FIGURE 1 Keenan at Fig. 1; Keenan Prov at 2.1.
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; Keenan Prov at 1. “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; Keenan Prov at 2. “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; Keenan Prov at 33. “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 misconstruction of the diagram or process.” Keenan at 4:46-60; Keenan Prov at 34. “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; Keenan Prov at 34. “Task Sequence (Topology): Depicts tasks in process order sequence including tasks that require use of information assets.” Keenan at 5:17-19; Keenan Prov at 35.
--	--

	“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; Keenan Prov at 3. “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; Keenan Prov at 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; Keenan Prov at 26-27. “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; Keenan Prov at 4-5. “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; Keenan Prov at 9. “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; Keenan Prov at 11. “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; Keenan Prov at 12-13. “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; Keenan Prov at 25. “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; Keenan Prov at 31-32. “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; Keenan Prov at 26-27. “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; Keenan Prov at 27. “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; Keenan Prov at 28. “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 Prov at 32.

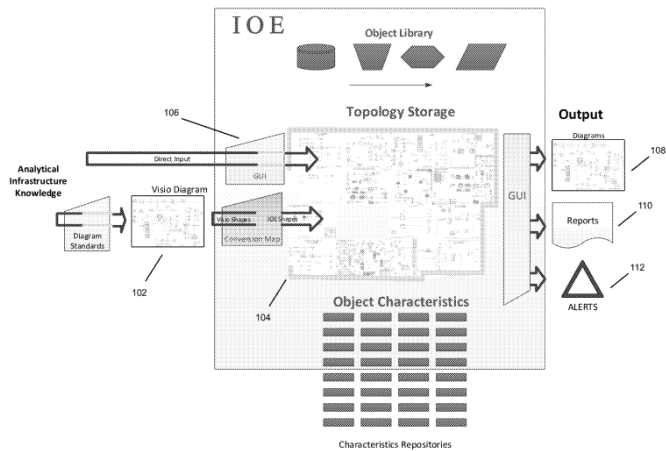


FIGURE 1

Keenan at Fig. 1; Keenan Prov at 2.1.

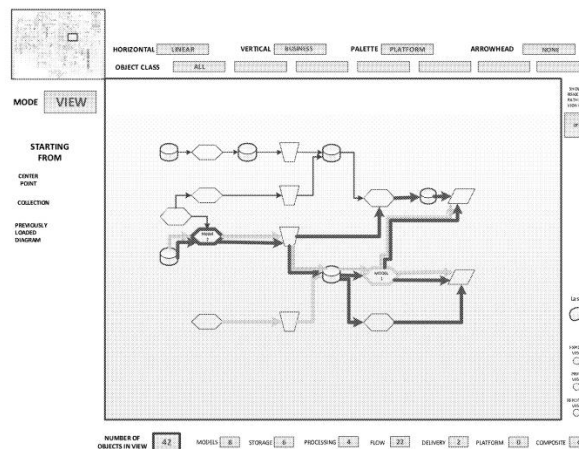


FIGURE 10

Keenan at Fig. 10; Keenan Prov at Fig. 4.7.

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

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

interfaces, a threat report displaying each threat that is associated with 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; Keenan Prov at 3.

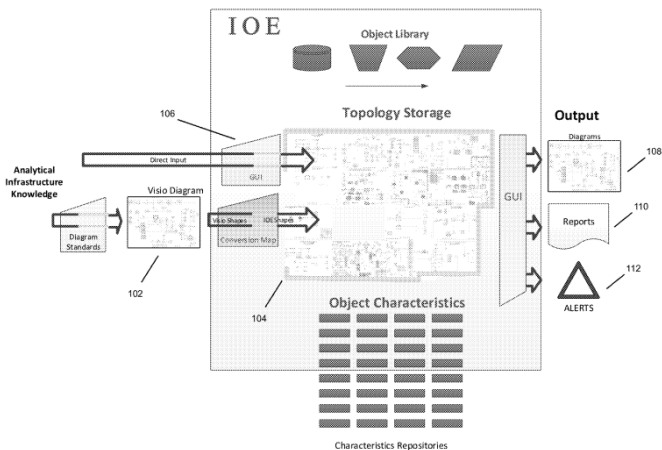


FIGURE 1

Keenan at Fig. 1; Keenan Prov at 2.1.

“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; Keenan Prov at 6.

“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; Keenan Prov at 22-24. "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; Keenan Prov at 23. "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; Keenan Prov at 25. "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; Keenan Prov at 29-30.

“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; Keenan Prov at 30.

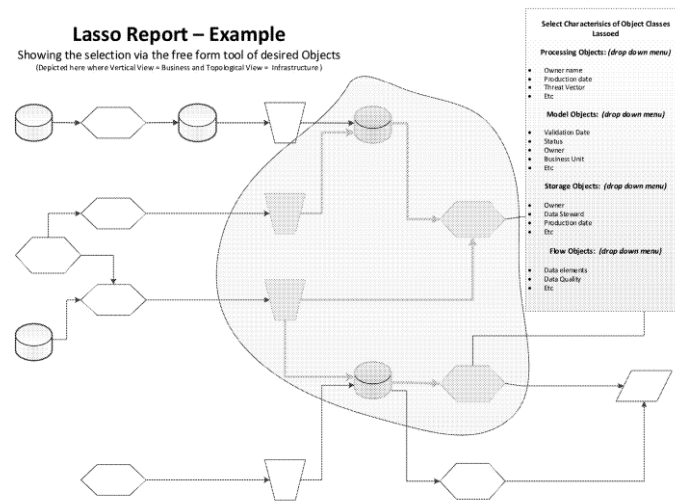
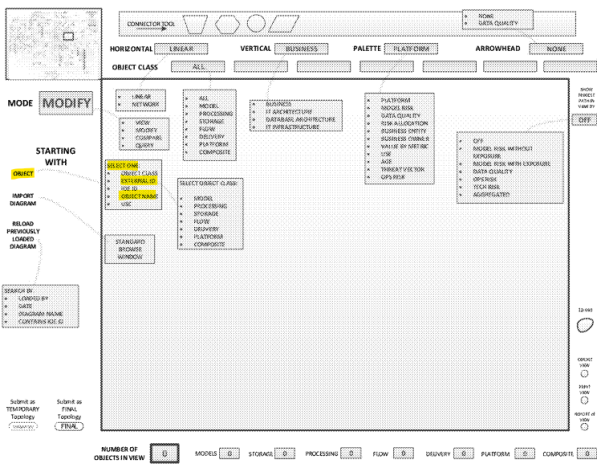


FIGURE 16

	Keenan at Fig. 16; Keenan Prov at Fig. 6.3.
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; Keenan Prov at 3, 17 (Table 4.1), Fig. 4.3.
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; Keenan Prov at 27. “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; Keenan Prov at 6. “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; Keenan Prov at 16.
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; Keenan Prov at 26-27. 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	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

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.	(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 6:13-26; Keenan Prov at 4. “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; Keenan Prov at 26-27. “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; Keenan Prov at 27.  FIGURE 13 Keenan at Fig. 13; Keenan Prov at Fig. 5.2.
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; Keenan Prov at 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; Keenan Prov at 28. “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; Keenan Prov at 16. "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; Keenan Prov at 26-27. "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 Prov at 27.
--	--

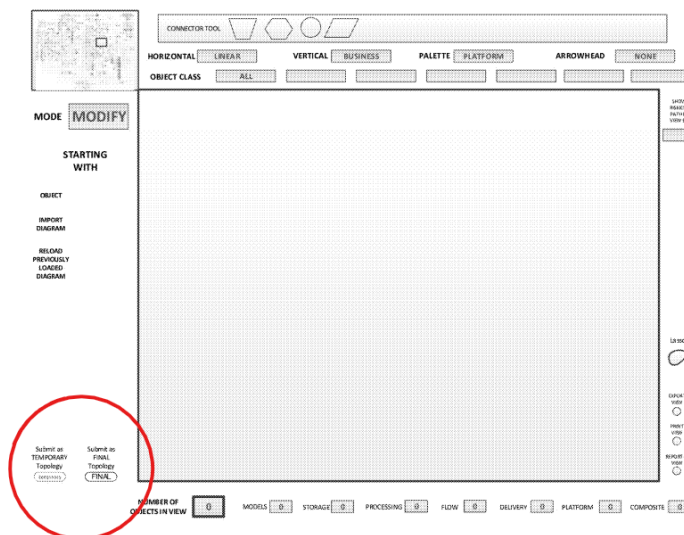


FIGURE 12

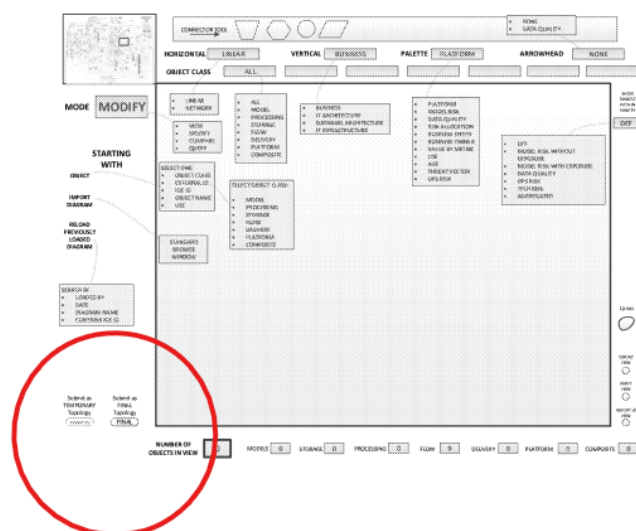


FIGURE 13

Keenan at Fig. 12-13; Keenan Prov at Figs. 5.1, 5.2.

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; Keenan Prov at 26-27.

“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; Keenan Prov at 1. “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; Keenan Prov at 34. “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 misconstruction of the diagram or process.” Keenan at 4:54-60; Keenan Prov at 34. “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; Keenan Prov at 3. “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; Keenan Prov at 5. “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; Keenan Prov at 6-7. “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; Keenan Prov at 18. “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; Keenan Prov at 26-27.
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; Keenan Prov at 3, 17 (Table 4.1), Fig. 4.3. 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 threat model components	See Element 1[b].

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; Keenan Prov at 1. “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 Prov at 26-27. “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; Keenan Prov at 25. “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 Prov at 26-27.
--	---

	“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; Keenan Prov at 31-32. “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; Keenan Prov at 26-27. “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; Keenan Prov at 28. “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 Prov at 32.

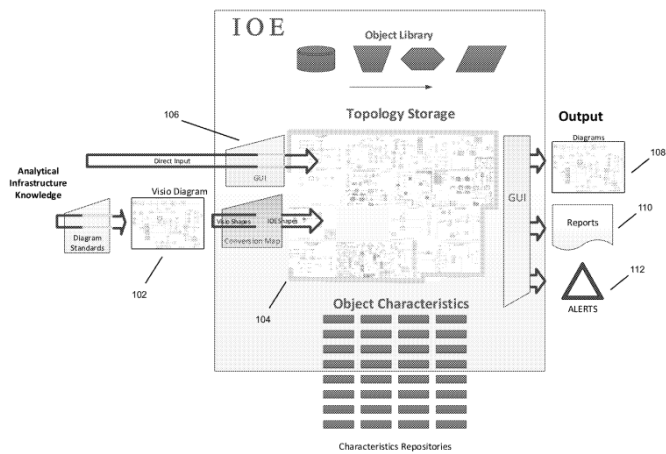


FIGURE 1

Keenan at Fig. 1; Keenan Prov at 2.1.

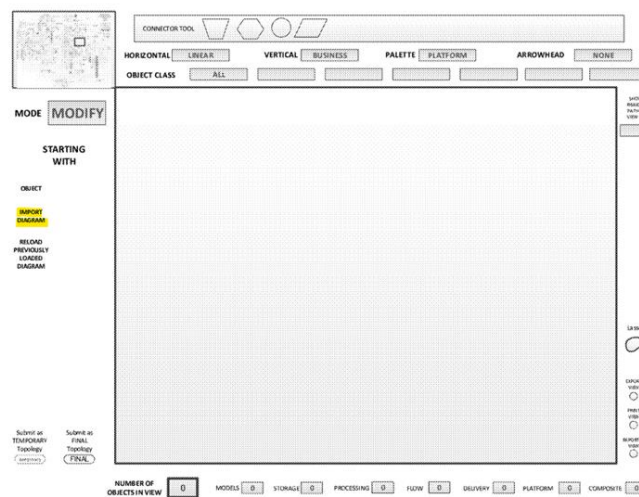
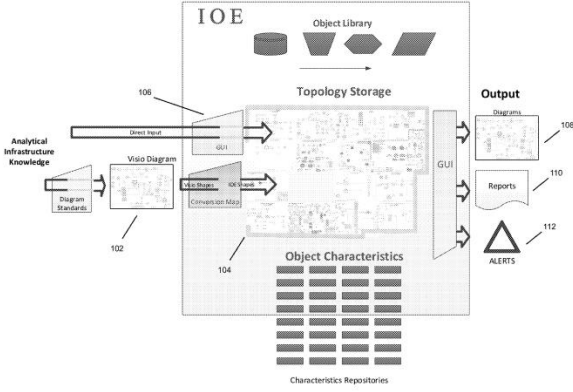
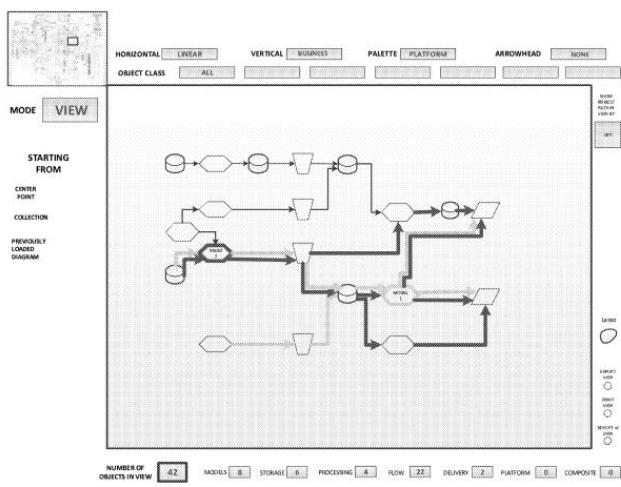


FIGURE 12

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; Keenan Prov at 12-13. 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; Keenan Prov at 2. “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; Keenan Prov at 16.

	 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.	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; Keenan Prov at 3.
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	See Claim 2.

the one or more computing devices.	
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 file, a PNG file, and a JPEG file.	See Claim 4.
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; Keenan Prov at 4. “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 Prov at 27.

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; Keenan Prov at 4. “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; Keenan Prov at 9.
--	---

	“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; Keenan Prov at 24. “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; Keenan Prov at 26-27. “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 Prov at 27.
--	--

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 input, modify the one or more mapping files.	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 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’; Keenan Prov at 1. “Canvas: Physical screen space in the IOE Modification layer where user builds or modifies topology” Keenan at 4:1-2; Keenan Prov at 33. “2) a Modify mode, where the user can add, change and delete Objects or Characteristics;” Keenan at 11:46-47; Keenan Prov at 25-26. “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; Keenan Prov at 13. “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; Keenan Prov at 31-32. “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; Keenan Prov at 28. “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; Keenan Prov at 31-32. “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; Keenan Prov at 26-27.
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 through the one or more databases;	See Element 1[a] and 8[a].
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	See Element 8[c].

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;	
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	See Element 1[i] and 8[f].

databases with one of the threat model components included in the threat model.	
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.