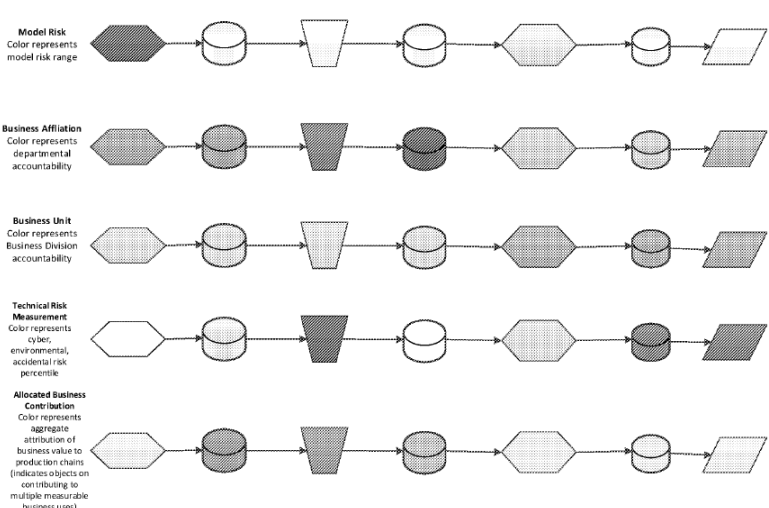
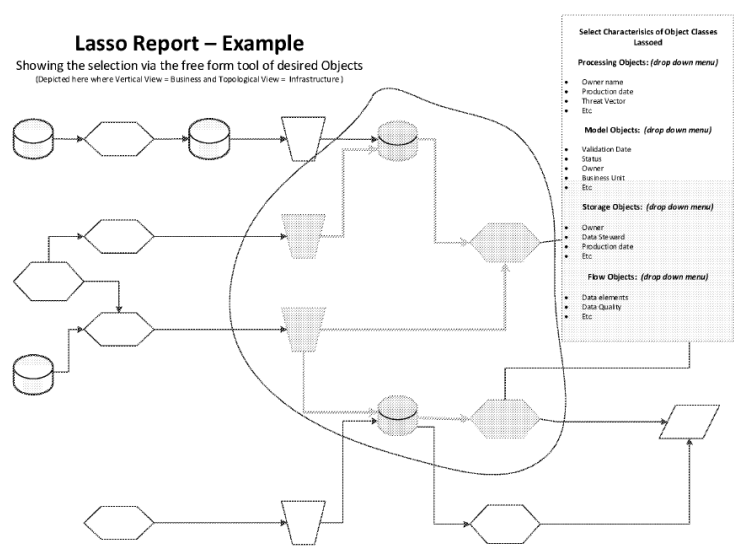


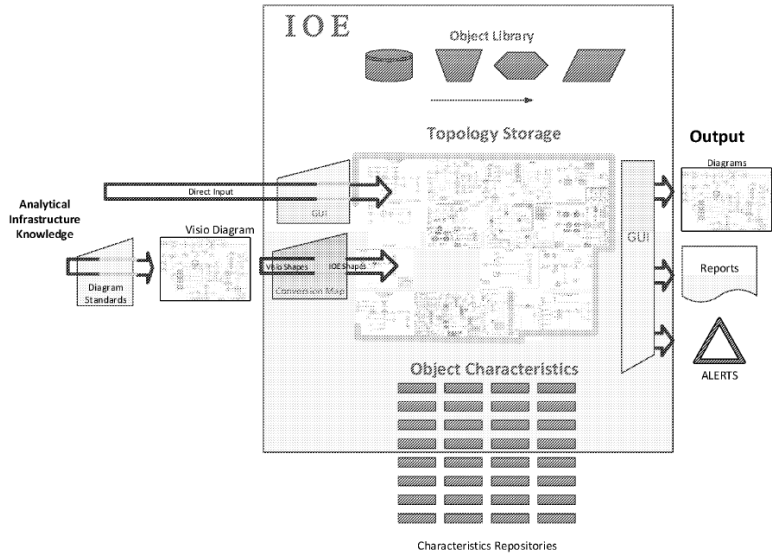
Comparison of '366 Claims Against Keenan Provisional

10,713,366 “Systems and methods for automated threat model generation from third party diagram files”	The '366 Patent is Anticipated by US Patent Application No. 62/487,370 (“Keenan Provisional”)
1[p]. A threat modeling method, comprising:	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional 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. 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. p. 1. 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. 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. p. 1.

	Figure 4.4 Examples of Palette View Palette View – Examples Same topology depicted using different Palette Views (Depicted here where Vertical View = Business and Topological View = Infrastructure)  Model Risk Color represents model risk range Business Affiliation Color represents departmental accountability Business Unit Color represents Business Division accountability Technical Risk Measurement Color represents cyber, environmental, accidental risk percentile Allocated Business Contribution Color represents aggregate attribution of business value to production chains (indicates objects on contributing to multiple measurable business uses) Figure 4.4. Figure 6.3 Lasso Report – example of selecting objects using the Lasso function 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) - Owner name - Production date - Threat Vector - Etc Model Objects: (drop down menu) - Validation Date - Status - Owner - Business Unit - Etc Storage Objects: (drop down menu) - Owner - Data Steward - Production date - Etc Flow Objects: (drop down menu) - Data elements - Data Quality - Etc Figure 6.3.
1[a]. providing one or more databases, the one or more databases comprising:	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional 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. p. 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 Provisional discloses this claim limitation. For example, Keenan Provisional 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. 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. p. 1. 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. 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. p. 1.
--	---

	Figure 4.4 Examples of Palette View Palette View – Examples Same topology depicted using different Palette Views (Depicted here where Vertical View = Business and Topological View = Infrastructure) Figure 4.4. Figure 6.3 Lasso Report – example of selecting objects using the Lasso function Lasso Report – Example Showing the selection via the free form tool of desired Objects (Depicted here where Vertical View = Business and Topological View = Infrastructure) Figure 6.3.
1[c]. providing one or more mapping files communicatively coupled with the one or more databases,	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3.

	A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1  Figure 2.1.
1[d]. the one or more mapping files correlating the threat model components with visual diagram components of a third party software application (hereinafter	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3.

“third party diagram components”); and

A. Import 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 object already exists:

A message is posted to the screen “The following objects matched existing objects from The IOE, new relationships will be loaded to existing object.

The existing Objects will be updated with new relationships based on the imported visio diagram.

The import function also provides the following utilities

Provide object count by class and total count once diagram is imported.

The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above)

pp. 26-27.

B. Updating Characteristics

The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface.

p. 32.

Figure 2.1

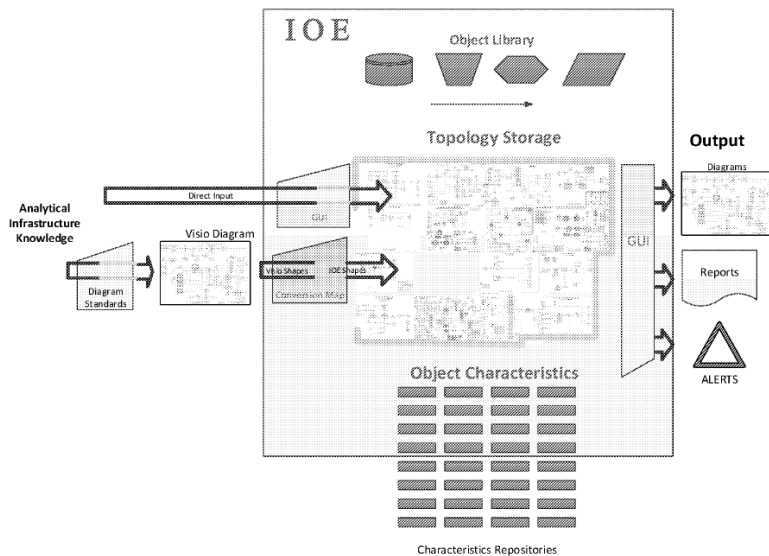


Figure 2.1.

1[e]. in response to receiving one or more user inputs, using one or more user interfaces displayed on one or more computing devices communicatively

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

The system provides a wide-variety of visual presentation options of the topology allowing the user to view, using GUI, a subset of the topology by selecting use, object function class, functional specificity, relationship to specific object and other criteria. It provides a medium for applying user-provided logic to the topology directly, so that any component or aspect of it can be isolated, retrieved, and assessed.

coupled with the one or more databases:	The system preferably operates according to the following principles: • Every object class is unique and designed to allow for specific features including states. • A library of object classes created is parsimonious to allow for a wide range of uses and allows for a high degree of scalability and modularization. • The IOE stores objects and their relationships as well as the object's structural metadata ("Characteristics"). • 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. • There is a one to one relationship between an object class and a diagram shape. • The diagramming standards are pre-specified sets of object shapes and formats and other visual elements. • Diagrams created in the IOE will 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. • The IOE preferably differentiates between 'hard data' (i.e. observations and measurements, and arithmetic combinations thereof) and 'calculations' which represent modeled combinations of inputs. We refer to the former only as 'data' and the latter as 'calcs'. • Read and write access is for multiple levels of engagement: A. Access and permissions are incorporated based on user function, structural metadata (Characteristics) describing business organization structure and business use cases. B. All user levels have access to a simple, fast, and feature-rich application set. • At least one object in a Task Sequence topology exists in an Infrastructure topology. pp. 1-2. III. Graphical User Interface There are four modes for the user to interact with the IOE via a Graphical User Interface (GUI). View: View any section of the topology and Characteristics by Object Modify: Add, change and delete Objects or Characteristics Compare: Visualize the difference between two diagrams
--	--

	Query: 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 slight different screen format. We refer to the portion of the GUI that the user sees as the “screen” as opposed to the logic which dictates what appears on the screen which would be the unseen parts of the GUI. Figure 3.View model Figure 3.2 View mode showing drop down values for user-selectable boxes The screen contains the following basic components View Box: This component uses most of the screen. Within the View Box the user sees the topology to view, modify or compare. It is view-only in the View and Compare modes and interactive in the Modify mode. Mini Map: In the upper left corner is a depiction of the topology. The default view is the entire company. Of course objects and labels are far too small to be seen individually but the purpose of the mini map is to show the user where within the entire company topology the current View is located. This would be analogous to having a street map of Altoona, PA in the main part of the screen (the View Box) while the Mini Map would indicate where on a map of the US Altoona is. The Mini Map defaults to the following View options • Horizontal: LINEAR • Vertical: BUSINESS • Palette: PLATFORM • Arrowhead: NONE • Object Class: ALL
--	---

Locator Box: Within the Mini Map is the Locator Box. Regardless of how the user zooms in and out of the Mini Map, the Locator Box will encompass the same topology and so will expand or contract to remain in sync with the Mini Map display as the zoom function is exercised. When the cursor is not in the Mini Map the Locator Box will display the same boundaries represented by the View in the View Box. It does this by automatically shifting position, shrinking and expanding borders to encompass topology represented in the View Box.

Table 3.1 MINI MAP FUNCTIONALITY

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.

pp. 12-14.

Figure 3.1 View model

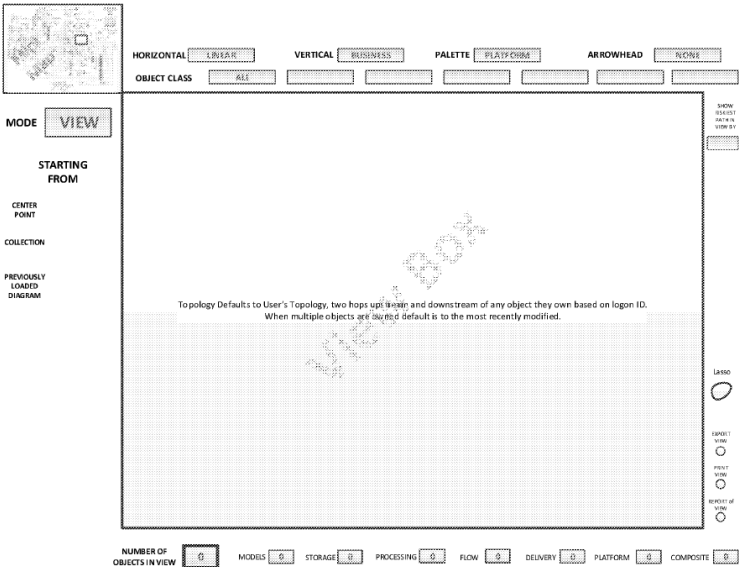


Figure 3.1.

Figure 3.2 View mode showing drop down values for user-selectable boxes

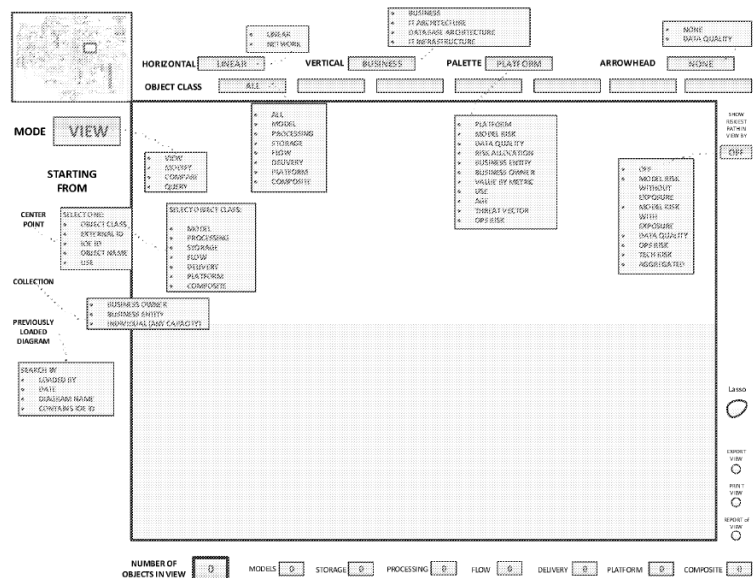


Figure 3.2.

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

II. Core Technology

The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1

p. 3.

A. Import 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 object already exists:

A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object.

The existing Objects will be updated with new relationships based on the imported visio diagram.

The import function also provides the following utilities

Provide object count by class and total count once diagram is imported.

The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above)

pp. 26-27.

	B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1 The diagram illustrates the IOE (Infrastructure Object Engine) architecture. At the top, 'Analytical Infrastructure Knowledge' (represented by a cylinder) feeds into 'Diagram Standards' (a trapezoid), which then feeds into a 'Visio Diagram' (a rectangle). The 'Visio Diagram' is processed by 'Visio Shapes' (a trapezoid) and a 'Conversion Map' (a rectangle) to produce 'IOE Shapes' (a trapezoid). These 'IOE Shapes' are then imported into the 'IOE' (a large rectangle). The 'IOE' also receives 'Direct Input' (a rectangle) and 'GUI' (a rectangle) inputs. Inside the 'IOE', there is an 'Object Library' (a cylinder) and 'Topology Storage' (a rectangle). The 'IOE' outputs 'Object Characteristics' (a large rectangle) and 'Characteristics Repositories' (a grid of rectangles). The 'IOE' also outputs 'Output' (a rectangle), which includes 'Diagrams' (a rectangle), 'Reports' (a rectangle), and 'Alerts' (a triangle). Figure 2.1.
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 Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3. A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27.

	B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1 Figure 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 Provisional discloses this claim limitation. For example, Keenan Provisional discloses: The system provides a wide-variety of visual presentation options of the topology allowing the user to view, using GUI, a subset of the topology by selecting use, object function class, functional specificity, relationship to specific object and other criteria. It provides a medium for applying user-provided logic to the topology directly, so that any component or aspect of it can be isolated, retrieved, and assessed.

	The system preferably operates according to the following principles: • Every object class is unique and designed to allow for specific features including states. • A library of object classes created is parsimonious to allow for a wide range of uses and allows for a high degree of scalability and modularization. • The IOE stores objects and their relationships as well as the object's structural metadata ("Characteristics"). • 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. • There is a one to one relationship between an object class and a diagram shape. • The diagramming standards are pre-specified sets of object shapes and formats and other visual elements. • Diagrams created in the IOE will 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. • The IOE preferably differentiates between 'hard data' (i.e. observations and measurements, and arithmetic combinations thereof) and 'calculations' which represent modeled combinations of inputs. We refer to the former only as 'data' and the latter as 'calcs'. • Read and write access is for multiple levels of engagement: A. Access and permissions are incorporated based on user function, structural metadata (Characteristics) describing business organization structure and business use cases. B. All user levels have access to a simple, fast, and feature-rich application set. • At least one object in a Task Sequence topology exists in an Infrastructure topology. pp. 1-2. III. Graphical User Interface There are four modes for the user to interact with the IOE via a Graphical User Interface (GUI). View: View any section of the topology and Characteristics by Object Modify: Add, change and delete Objects or Characteristics Compare: Visualize the difference between two diagrams
--	--

	Query: 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 slight different screen format. We refer to the portion of the GUI that the user sees as the “screen” as opposed to the logic which dictates what appears on the screen which would be the unseen parts of the GUI. Figure 3.View model Figure 3.2 View mode showing drop down values for user-selectable boxes The screen contains the following basic components View Box: This component uses most of the screen. Within the View Box the user sees the topology to view, modify or compare. It is view-only in the View and Compare modes and interactive in the Modify mode. Mini Map: In the upper left corner is a depiction of the topology. The default view is the entire company. Of course objects and labels are far too small to be seen individually but the purpose of the mini map is to show the user where within the entire company topology the current View is located. This would be analogous to having a street map of Altoona, PA in the main part of the screen (the View Box) while the Mini Map would indicate where on a map of the US Altoona is. The Mini Map defaults to the following View options • Horizontal: LINEAR • Vertical: BUSINESS • Palette: PLATFORM • Arrowhead: NONE • Object Class: ALL
--	---

Locator Box: Within the Mini Map is the Locator Box. Regardless of how the user zooms in and out of the Mini Map, the Locator Box will encompass the same topology and so will expand or contract to remain in sync with the Mini Map display as the zoom function is exercised. When the cursor is not in the Mini Map the Locator Box will display the same boundaries represented by the View in the View Box. It does this by automatically shifting position, shrinking and expanding borders to encompass topology represented in the View Box.

Table 3.1 MINI MAP FUNCTIONALITY

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.

pp. 12-14.

Figure 3.1 View model

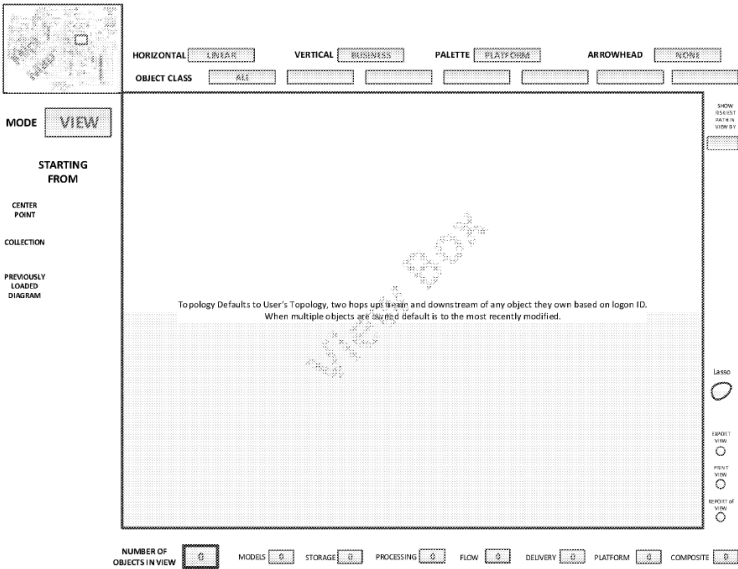


Figure 3.1.

Figure 3.2 View mode showing drop down values for user-selectable boxes

The screenshot shows a complex user interface for viewing a topology diagram. At the top, there are several dropdown menus for filtering the view: 'HORIZONTAL' (with options: LINEAR, NETWORK), 'VERTICAL' (with options: BUSINESS, PALETTE, PLAT FORM), and 'ARROWHEAD' (with options: NONE, DATA QUALITY). Below these are 'OBJECT CLASS' and 'MODE' dropdowns. The 'MODE' dropdown is currently set to 'VIEW'. On the left side, there are sections for 'STARTING FROM' (with options: SELECTED, SELECTED CLASS, SELECTED TO, SELECTED FROM, SELECTED TO, SELECTED FROM), 'CENTER POINT' (with options: SELECTED, SELECTED CLASS, SELECTED TO, SELECTED FROM, SELECTED TO, SELECTED FROM), and 'PREVIOUSLY LOADED DIAGRAM' (with options: SELECTED, SELECTED CLASS, SELECTED TO, SELECTED FROM, SELECTED TO, SELECTED FROM). The main area is a large grid displaying a topology diagram. At the bottom, there are several status indicators: 'NUMBER OF OBJECTS IN VIEW' (0), 'MODELS' (0), 'STORAGE' (0), 'PROCESSING' (0), 'FLOW' (0), 'DELIVERY' (0), 'PLATFORM' (0), and 'COMPOSITE' (0). On the right side, there are additional dropdown menus for 'VIEW BY' (with options: NONE, DATA QUALITY) and 'VIEW OF' (with options: NONE, DATA QUALITY).

Figure 3.2.

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

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

C. Alerts

Informing contributors and consumers of the Topology is a key benefit of the IOE. Since characteristics are linked to objects in the topology the changing of key characteristics affects not only the contributors and consumers of that object but also those contributors and consumers downstream. Alerts can be built that can take advantage of the community and accountability within a common topology to warn

others of changes that impact them by virtue of their location on the topology. There are two basic types of Alerts.

1. Key Characteristic Changes:

Conditions can be set for each object class for Alerts to be generated upon the change in a characteristic when a new value is entered that outside set parameters. These values can either be associated with the object in the binary object code or can be characteristics in the relevant repository of characteristics for that object class. A period delta check of the repository could look for new values outside the parameters for each object class.

The Alerts can be tailored based on the relationship with Upstream and Downstream objects. The Alerts will be sent via email or other means of electronic communication to advise the owner of the targeted object that a change to a characteristic with a related object has changed outside parameters. An example of this would be the change to model status. When a model validation expires this needs to be communicated to downstream objects of the model. An email would be generated to the owners of the model classes deemed to be impacted by the change to the model characteristic of validation status to inform them of this fact so that they may take action to mitigate the resulting risk.

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 can be 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.

2. Object Capability Changes:

A more robust set of Alerts can be used to notify upstream and downstream object owners that an object has been added/removed/replaced or otherwise undergone alteration in capabilities. This Alert would be signaled as higher importance than a Characteristic change.

An example would be where a model owner sources data from a new vendor. This addition to the IOE would then trigger an alert to all downstream object owners.

3. New use of sourced data and generated calculations:

When users draw flow objects from storage objects and add data elements to the Flow Object Characteristics the IOE will locate the source of the calculations (Model Object) and the source of Data Elements (Storage Object). Once identified, the IOE will send an Alert to the owner of the Model and Storage object advising them that their Data and Calculations are being sent to a new object.

pp. 30-31.

Figure 2.1

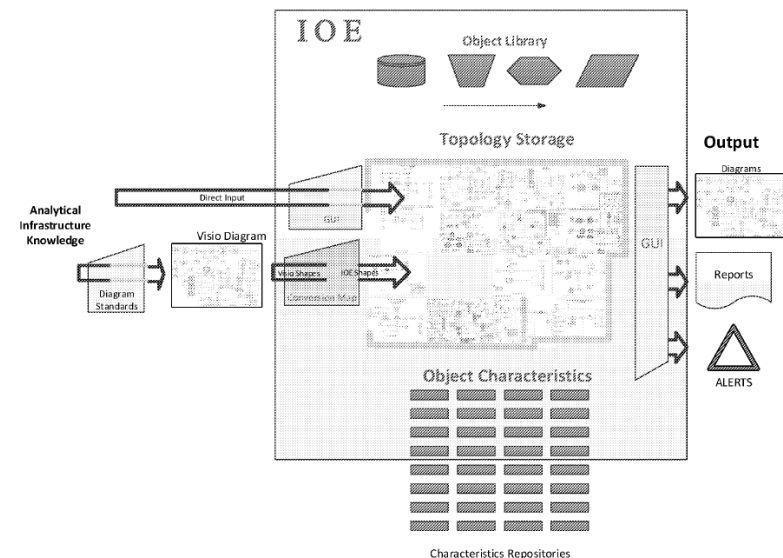


Figure 2.1.

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

II. Core Technology

The IOE is intended to be effectively a dynamic map of the firm’s analytic infrastructure 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 Figure 2.1

p. 3.

A. Import 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 object already exists:

A message is posted to the screen “The following objects matched existing objects from The IOE, new relationships will be loaded to existing object.

The existing Objects will be updated with new relationships based on the imported visio diagram.

The import function also provides the following utilities

Provide object count by class and total count once diagram is imported.

The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above)

pp. 26-27.

B. Updating Characteristics

The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface.

p. 32.

Figure 2.1

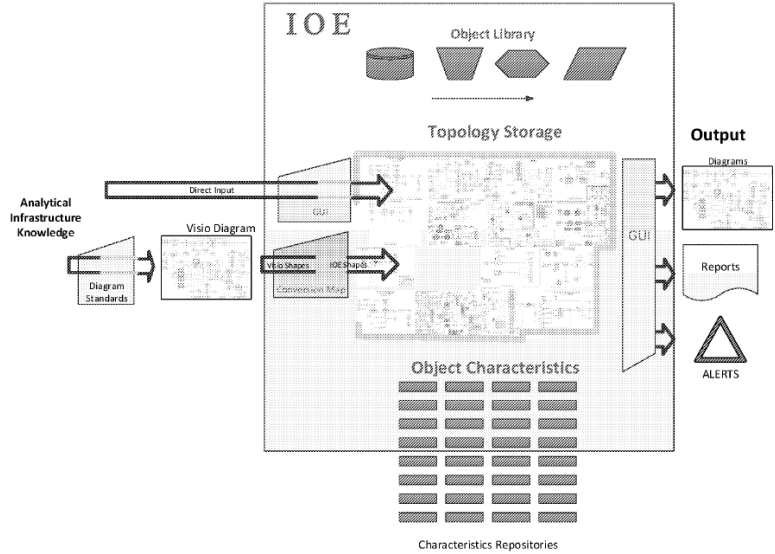


Figure 2.1.

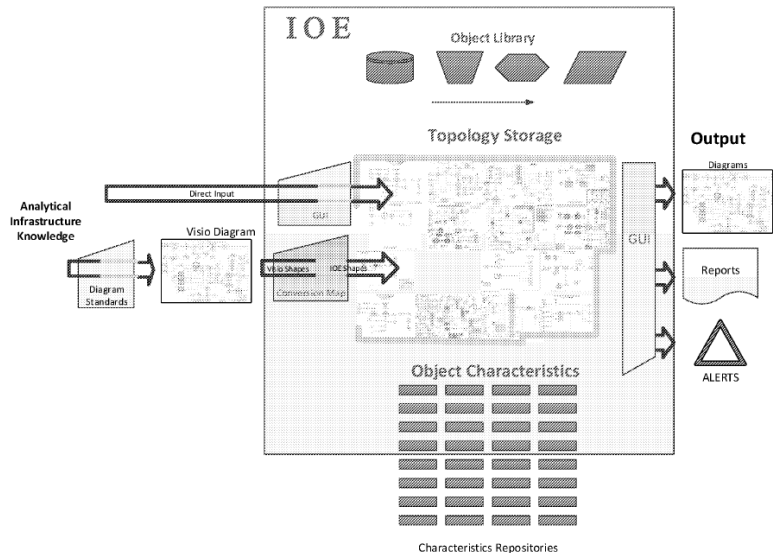
3. The method of claim 1, wherein the one or more mapping files comprise one or more tables stored within the one or more databases.	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3. A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1 
---	---

	Figure 2.1.
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 Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3. A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1

	Figure 2.1.
5. The method of claim 1, wherein determining, for the subset of third party diagram components, correlated threat model components, comprises first determining whether a user-defined name for one of the subset of third party diagram components has a match in the one or more mapping files and, if no match is found, then determining whether a default title for the third party diagram component, as defined by the third party software application, has a match in the one or more mapping files.	Keenan Provisional discloses, or at least renders obvious, this claim limitation. For example, Keenan Provisional discloses: A. Object Classes Each object class and sub-class is itself represented as a list of fixed characteristics – that fixed set of characteristics define the object class, and establish the response mode for the logical functions (described below). The object class may also have a state variable as part of the characteristic set. Each class may also permit extension to include user-defined characteristics, however from a core technology perspective such user-defined characteristics should not affect the behavior of the objects. Objects that contain all of the information for the fixed characteristic set that defines them are considered “fully identified”. Objects that do not contain all of the information for the fixed characteristic set that defines them are considered “under-identified”. Under-identified objects may exist with the IOE but may not be supported by the full complement of internal functions. Each object should have 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 need to carry their repository IDs as well as an ID internal to the IOE. By extension then, all objects in the IOE, with the exception of data flow objects must be 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. Every object in the IOE should carry with it, in addition to its defining object-specific and topology-related information: i) An object class identifier ii) A creation date iii) A creation source code iv) An internal IOE ID v) An external repository ID Objects independent of the infrastructure can be stored in the IOE. These relationships are referred to as Task Sequence relationships as opposed to Infrastructure relationships. The analytical infrastructure is a network of analytical components such as models, EUTs, databases, reports and dataflow information assets. There is immense diversity in the ecosystem of the analytical infrastructure. However, the IOE framework and approach is to place structure within this diversity and group the real world components into object classes. Objects classes can be further grouped into sub-classes. Objects should be as parsimoniously constructed as possible to enhance control, reduce clutter and maintenance costs. For each object class, characteristics are grouped into three categories: i) Required (mandatory, failure to include results in an under-identified object) ii) Pre-specified (information normally expected to, that supports general querying), and iii) User-defined (allowable extensions to the data/metadata embedded directly in the object) The following are Object Classes: 1. Storage Objects Data Storage object class includes databases, text files, spreadsheets wherein the purpose is to store data – even if temporarily. Effectively, if data, destined for inclusion in analysis or reporting, is held in a location a data storage object must exist. While we view data as “physical” and part of the overall analytic infrastructure, we do not seek to objectify production data at rest and in bulk. . For data

storage objects, certain boundary cases can be identified: models and modeling platforms usually require input files, which may remain resident as objects serving only within those environments. As these tend to be single-use objects and not formatted for secondary retrieval, we would exclude them from the topology. Alternatively, certain modeling platforms require the creation of data files that do contemplate re-use or multipurpose use, and explicitly facilitate their use for data storage. SAS data files/databases, for example, fall into this category and should be included in the topology as data storage objects.

Such a broad definition of data storage objects suggests that a great many may not be inventoried in any external repository, and key characteristics and metadata may not be available. However we think the trade-off is beneficial, and prioritize the clarity of the object classification over a reduction in under-identified objects. Very likely, under-identified data storage objects would fit primarily into a few sub-classes (e.g. non-model EUTs) and small improvements in the control environment could reduce their number significantly.

pp. 3-5.

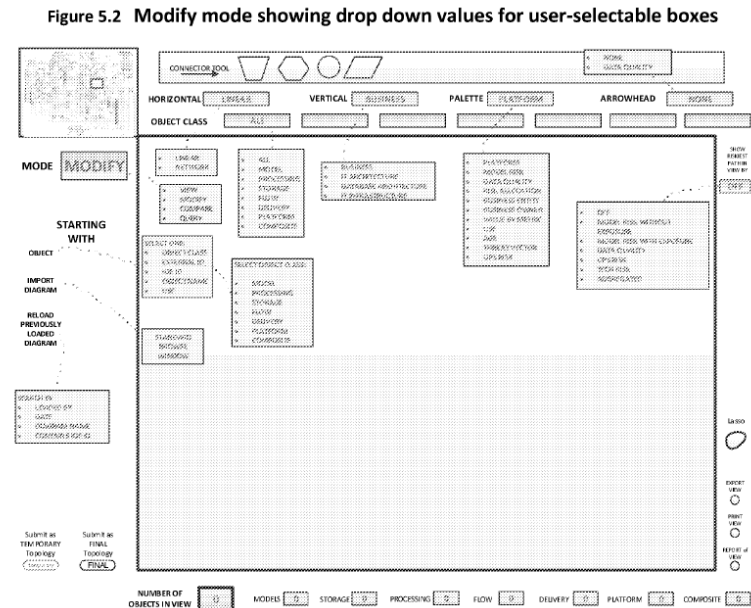


Figure 5.2.

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:

E. Saving Modifications

Unless saved as FINAL, this 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 person modifying can name the temporary saved topology. Saving as temporary – any changes between existing and temporary adds are then highlighted in the same manner as compare.

p. 28.

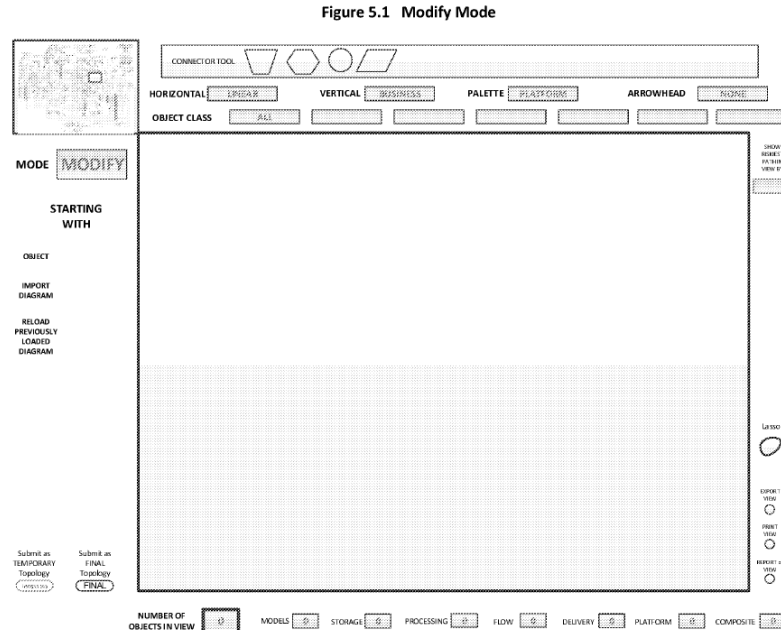


Figure 5.1.

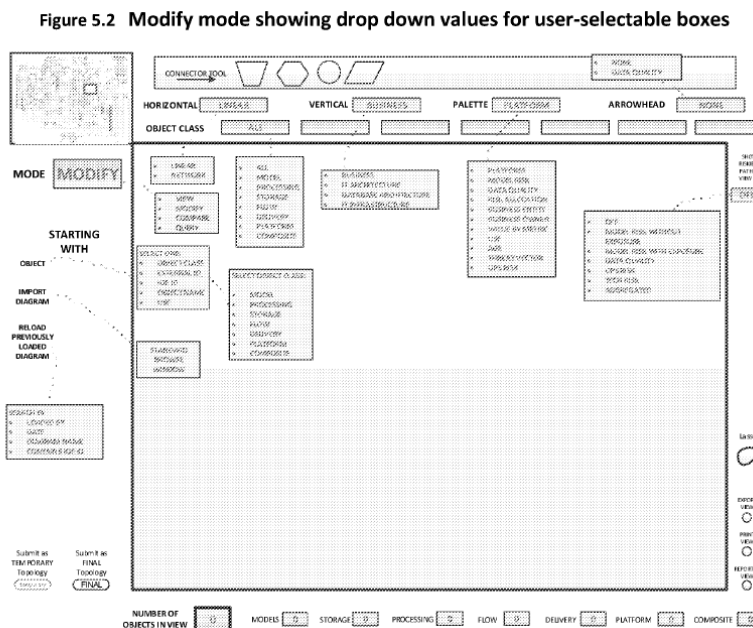


Figure 5.2.

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

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

II. Core Technology

The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1

as defined in the one or more mapping files.	p. 3. A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1 Figure 2.1.
8[p]. A threat modeling system, comprising: one or more computing devices communicatively coupled with one or more	See Elements 1[p]-1[a] and 1[e]. See also:

databases, the one or more databases comprising:	Figure 2.1 Figure 2.1.
8[a]. a plurality of threat model components stored therein; and a plurality of threats stored therein, wherein each of the threats is associated with at least one of the threat model components through the one or more databases;	See Element 1[b].
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	See Elements 1[e] and 1[f]. See also:

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;

II. Core Technology

The IOE is intended to be effectively a dynamic map of the firm’s analytic infrastructure 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 Figure 2.1

p. 3.

A. Import 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 object already exists:

A message is posted to the screen “The following objects matched existing objects from The IOE, new relationships will be loaded to existing object.

The existing Objects will be updated with new relationships based on the imported visio diagram.

The import function also provides the following utilities

Provide object count by class and total count once diagram is imported.

The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above)

pp. 26-27.

B. Updating Characteristics

The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface.

p. 32.

Figure 2.1

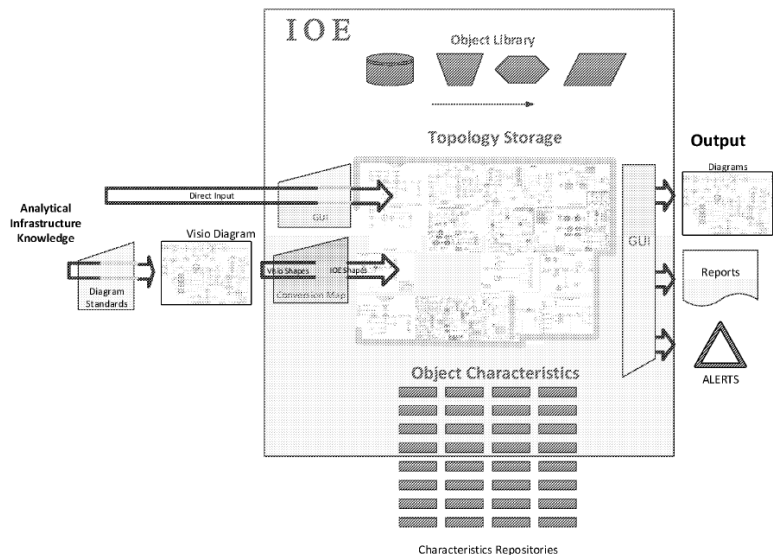
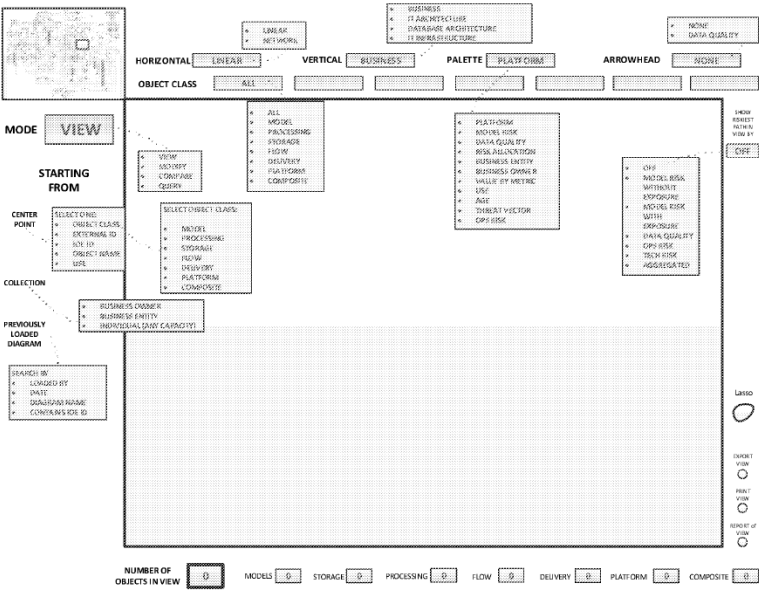
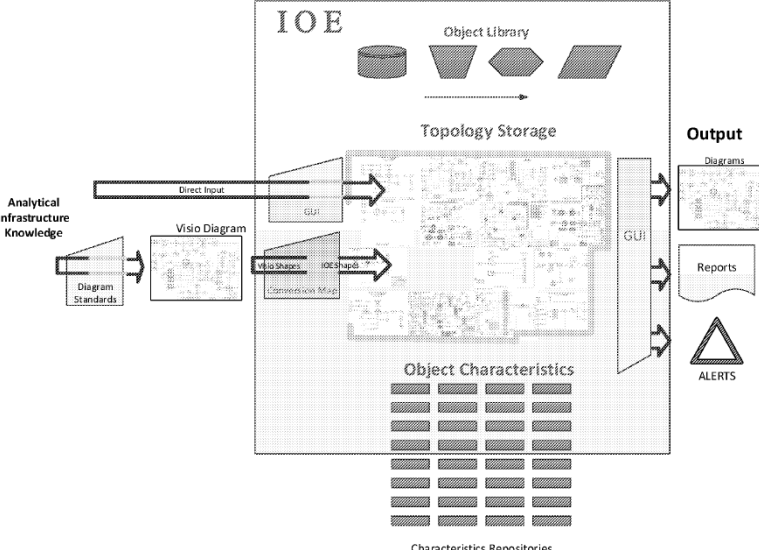


Figure 2.1.

8[d]. an interface configured to, in response

See Elements 1[e], 1[g], 8[c].

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;	
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	<i>See Element 1[h].</i> <i>See also:</i> 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. p. 2. III. Graphical User Interface There are four modes for the user to interact with the IOE via a Graphical User Interface (GUI). View: View any section of the topology and Characteristics by Object Modify: Add, change and delete Objects or Characteristics Compare: Visualize the difference between two diagrams Query: 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 slight different screen format. We refer to the portion of the GUI that the user sees as the “screen” as opposed to the logic which dictates what appears on the screen which would be the unseen parts of the GUI. pp. 12-13.

	Figure 3.2 View mode showing drop down values for user-selectable boxes  Figure 3.2.
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]. See also: Figure 2.1  Figure 2.1.
9. The system of claim 8, wherein the one or more mapping files are not stored in the one or more	See Claim 2.

databases but are stored on 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 Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3. A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32.

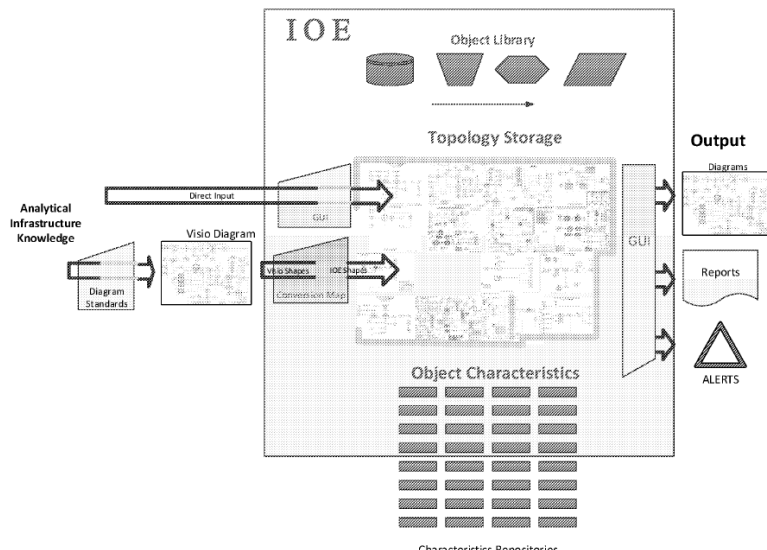
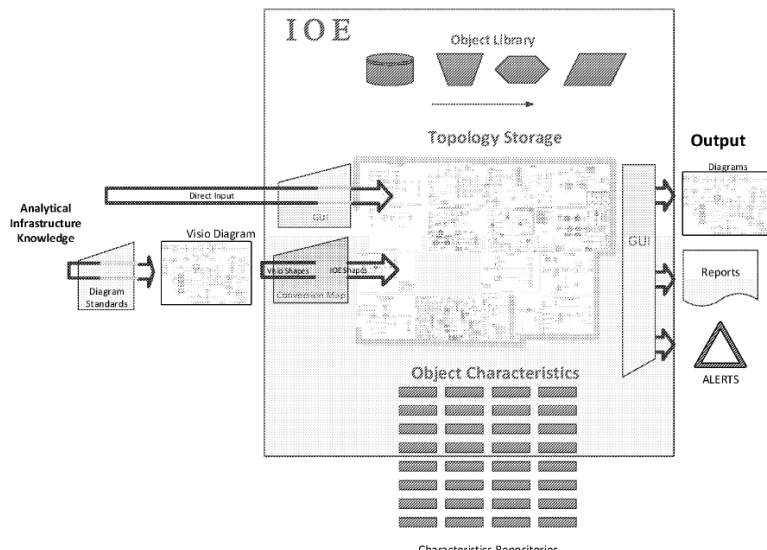
	Figure 2.1  Figure 2.1.
13. The system of claim 8, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.	Keenan Provisional discloses this claim limitation. For example, Keenan Provisional discloses: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm’s analytic infrastructure 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 Figure 2.1 p. 3. A. Import 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 object already exists: A message is posted to the screen “The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32.

	Figure 2.1  Figure 2.1.
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 Provisional discloses, or at least renders obvious, this claim limitation. For example, Keenan Provisional discloses: See Element 1[e]. See also: II. Core Technology The IOE is intended to be effectively a dynamic map of the firm's analytic infrastructure 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 Figure 2.1 p. 3.

	A. Import 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 object already exists: A message is posted to the screen "The following objects matched existing objects from The IOE, new relationships will be loaded to existing object. The existing Objects will be updated with new relationships based on the imported visio diagram. The import function also provides the following utilities Provide object count by class and total count once diagram is imported. The IOE will save the diagram objects so that it can be reproduced for audit purposes and to do diagram comparisons and allow users to select as a View in the future (see Selection of Topology to View above) pp. 26-27. B. Updating Characteristics The IOE is preferably informed of new and updated topological links and objects using importation of visio diagrams or by diagram maintenance interface. p. 32. Figure 2.1 Figure 2.1.
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 devices, the import interface configured to, in response to receiving user input, initiate reading of a data file by the one or more computing devices, the data file comprising a subset of the third party diagram components, the data file defining one or more relationships between the subset of third party diagram components;	See Element 8[c].
16[d]. an interface configured to, in response to receiving user input,	See Element 1[g] and 8[d].

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

19. The system of claim 16, wherein the one or more mapping files includes a file correlating user-defined names of the third party diagram components with the threat model components.	See Claim 12.
20. The system of claim 16, wherein the one or more mapping files includes a file correlating default titles of the third party diagram components with the threat model components.	See Claim 13.