Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2680978 C 2017/12/19

(11)(21) **2 680 978**

(12) BREVET CANADIEN / CANADIAN PATENT

(13) **C**

(22) **Date de dépôt/Filing Date:** 2009/10/02

(41) **Mise à la disp. pub./Open to Public Insp.:** 2010/04/08

(45) **Date de délivrance/Issue Date:** 2017/12/19

(30) **Priorité/Priority:** 2008/10/08 (US12/247,927)

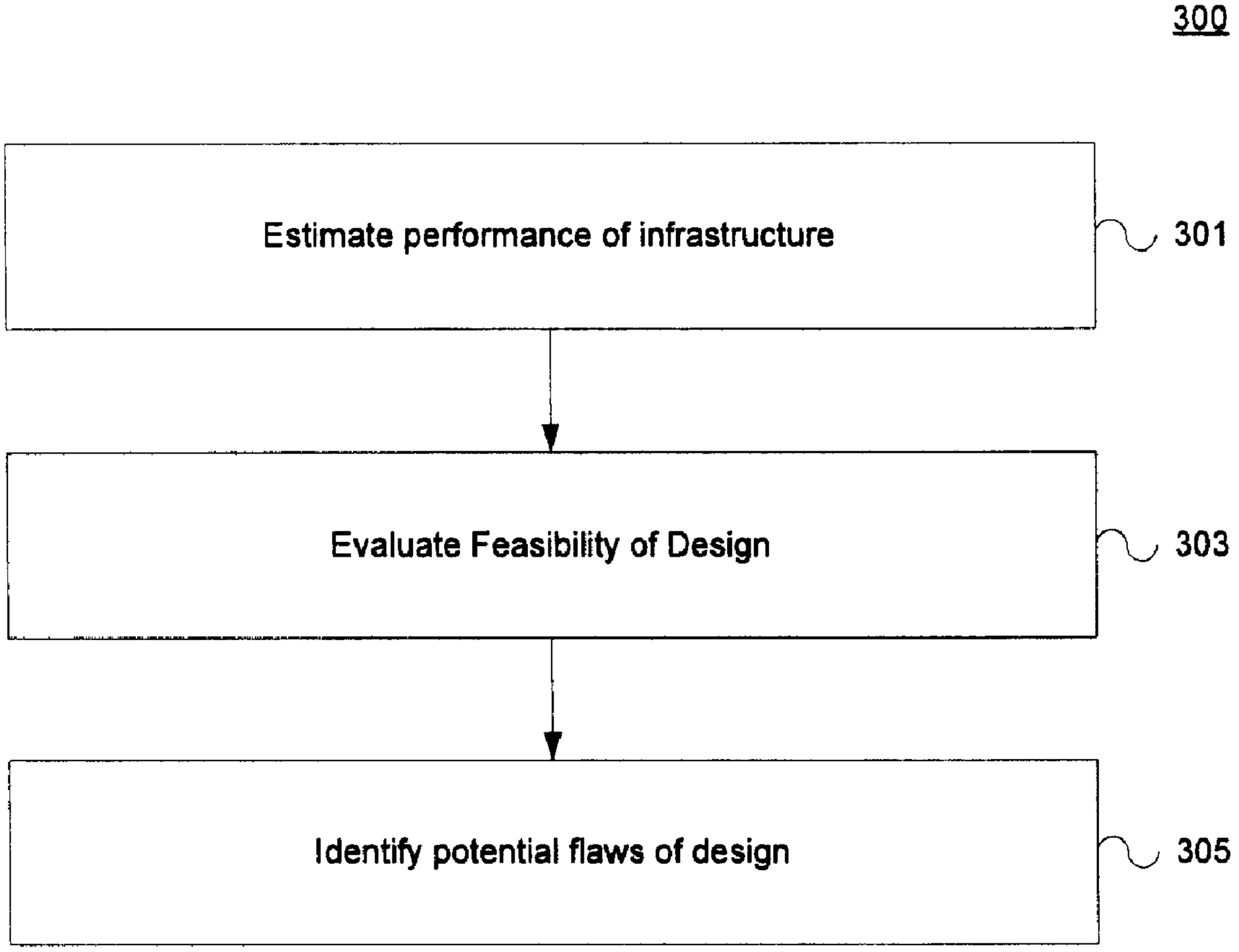
(51) **Cl.Int./Int.Cl.** ***G06F 9/44*** (2006.01),
G06F 17/30 (2006.01)

(72) **Inventeurs/Inventors:**
TUNG, TERESA, US;
LE, DANA, US;
VERMA, KUNAL, US

(73) **Propriétaire/Owner:**
ACCENTURE GLOBAL SERVICES LIMITED, IE

(74) **Agent:** SMART & BIGGAR

(54) **Titre : APPLICATION INTEGREE DE CONCEPTION**
(54) **Title: INTEGRATED DESIGN APPLICATION**



(57) **Abrégé/Abstract:**
Embodiments of the claimed subject matter provide a process and system for designing composite application systems. One embodiment of the claimed subject matter is a process of designing a composite application system to meet an application

CA 2680978 C 2017/12/19

(11)(21) **2 680 978**

(13) **C**

(57) **Abrégé(suite)/Abstract(continued):**
requirement. The process includes defining a use case scenario performed by a composite application system to meet the application requirement, integrating the use case scenario with an execution tool, and specifying an underlying infrastructure for the system design, wherein, the entire process is performed within an integrated software application.

INTEGRATED DESIGN APPLICATION

ABSTRACT

Embodiments of the claimed subject matter provide a process and system for designing composite application systems. One embodiment of the claimed subject matter is a process of designing a composite application system to meet an application requirement. The process includes defining a use case scenario performed by a composite application system to meet the application requirement, integrating the use case scenario with an execution tool, and specifying an underlying infrastructure for the system design, wherein, the entire process is performed within an integrated software application.

INTEGRATED DESIGN APPLICATION

BACKGROUND

[0001] The increasing popularity of businesses offering digital communication services (e.g. web services) have led to the development of robust, sophisticated systems capable of providing functionality for a diverse host of transactions. These systems are often conceived across one or more layers, typically including some combination of logical, hardware and process layers. Furthermore, these systems are frequently implemented as a composite of computerized hardware devices distributed across great distances - both logically and physically - all working in concert. Accordingly, the difficulty in designing these multi-layered, distributed systems to accurately and consistently achieve business objectives has grown to reflect the intricacy of meeting expanding demands while managing costs attributed to advancing system components.

[0002] The field of technology architecture has developed in response to address this need. A principle challenge in this particular field of architecture is to accurately capture a given set of objectives with one or more suitable business processes, and to implement the processes with the underlying physical infrastructure necessary to execute them successfully and efficiently. In large and complex systems, a single system design may include views of each layer of the system, e.g., the arrangement of system components, the processes

performed by the system in its execution, and the actual system components used. These systems are also referred to as "composite application systems."

[0003] As mentioned, a composite application system may be designed for one or more purposes and/or objectives. These purposes (and/or objectives) commonly include, for example, performing one or more actions, achieving one or more system goals, observing particular specifications, meeting certain application requirements, or some combination thereof. A design process typically includes ideating a solution for performing the one or more design purposes. An ideated solution represents an approach and/or design methodology to potentially achieve (or most reasonably approximate) the one or more purposes (and/or objectives) intended for the composite application system.

[0004] Naturally, the design process may involve the effort of multiple collaborators, each with their own areas of expertise. Unfortunately, misalignment between collaborators may occur, either due to conflicting interests, budget concerns or simple miscommunication. Misalignment between a business model and its underlying infrastructure in a design can lead to system inefficiency, sub-optimal performance, or worse, an abject failure to meet business goals. Valuable time may be spent debugging, communicating with other collaborators, and re-designing the system to correct misalignments. This in turn may lead to a lack of predictability in both cost and performance in the

system designed. In addition, the economic loss due to inefficiency compounded over the life of the system, or due to restructuring and re-provisioning system components can be enormous.

[0005] Misalignment in business system designs may result in poorly provisioned (either over-provisioned or under-provisioned) resources. Over-provisioning occurs when a system is designed with components having greater resources than what is actually needed to achieve application requirements, often at a higher cost than other suitable components or combinations and resulting in a waste of financial resources. In design projects with strict or limited budgets, over-provisioning one component may lead to budgetary restraints that restrict provisioning in other areas of the system to cheaper, less effective components that subsequently fail to operate at the level required by the design (under provisioning). Both over-provisioning as well as under-provisioning may also arise due to miscommunication of business objectives or a misunderstanding of system components.

[0006] Further amplifying the difficulty present in the current practice of technology architecture is the lack of both direct tool support and a unified standard of design. A lack of a direct tool and unified standard set may lead to collaborative misalignment, which in turn may result in inconsistent design quality, incompatible design formats and conflicting representations between and among regions, firms, individuals, or even design projects. Within a typical

design process, contributors to the design may utilize a host of existing applications and/or applications from various existing toolsets. Examples include applications specifically engineered for word processing, computer aided diagramming, email, spreadsheets, software configuration management and file transfer protocol.

[0007] An example of collaborative misalignment may include a representation of one component in a system design expressed by one collaborator as text in a document, whereas another collaborator may choose to represent another component as a diagram. Establishing a standard framework for every design project, and/or merging and conforming inconsistent representations may take up valuable time and coordinating effort. Furthermore, there is currently no well known or popular application that supports explicitly visualizing relationships across components and design views.

[0008] Furthermore, in systems with multiple collaborators where duplicates of the design are distributed among the collaborators, it may be difficult to identify the most current version of the system design. One or more of the collaborators may implement changes on locally stored copies, whereby subsequent manual merging of the modifications may be difficult and time-consuming, and automated merging of distributed, locally stored copies may risk losing some or all of the modifications.

[0009] Finally, a lack of standardized tool support also presents difficulties for efficiently reusing previous work products, since the lack of a unified standard often results in each system being conceived according to *ad hoc*, non-standard designs. While an experienced system designer may have a backlog of stored work that the designer may use as a template, the lack of a unified standard or tool may limit references to the designer's individual work. New and less experienced designers may lack access to such resources.

SUMMARY

[0010] This Summary is provided to introduce a selection of concepts in a simplified form that is further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

[0011] Embodiments are directed to an integrated design application. Specifically, a system for designing composite application systems. Embodiments address a visual tool for selecting and configuring hardware and software tools, which, when combined, form an information technology architecture. Functionality provided by various embodiments include (but are not limited to): providing users with multiple views of the processes included in a design; allowing users to explicitly create and view relationships across these views; allowing users to specify projected volumetric load with a beginning and end date; allowing users to specify the timescale (daily, hourly, weekly, etc) of those loads; enabling the assignment of corresponding target metrics to technical or logical processes; and enabling the identification of relationships within and among the infrastructure components whose estimated performance will correspond to the target metrics.

[0012] In one embodiment, a process is provided for designing a composite application system to meet one or more application requirements. According to the process, a composite application system is designed within a single software application. The composite application system is designed by defining one or more use case scenarios performed by an ideated composite application system to meet one or more application requirements. The application requirements are captured as a function of system load specified as volumetric information whose required output is specified as target metrics. Once the one or more use case scenarios are defined, the one or more use case scenarios are then integrated with one or more execution tools. An underlying infrastructure embodying the one or more execution tools are identified and selected to achieve system requirements.

[0012a] In an aspect, there is provided a process of designing a composite application system, the method comprising: defining a plurality of case scenarios performed by a composite application system in a composite application system design; integrating the plurality of case scenarios with a plurality of execution tools; specifying an underlying infrastructure for the system design to embody the plurality of execution tools to select a plurality of components whose combination meets a plurality of application requirements, wherein specifying the underlying infrastructure comprises: i) estimating a performance of the underlying infrastructure according to the composite application system design; ii) evaluating a feasibility of the composite application system design; and iii) identifying potential flaws of a feasible composite application system design, and

implementing the composite application system; wherein the process is performed in an integrated software application executed in a computing system.

[0012b] In another aspect, there is provided a computer-readable medium having computer executable instructions stored thereon for execution by one or more computers, that when executed performs a process of simulating the performance of a composite application system subject to a plurality of application requirements, the process comprising: receiving an input defining a plurality of case scenarios performed by a composite application system in a system design; integrating the plurality of case scenarios with a plurality of components of the composite application system, the plurality of components comprising a collection of software applications and hardware components; receiving an input specifying an underlying infrastructure for the system design to embody the plurality of components, wherein specifying the underlying infrastructure includes specifically selecting a plurality of software applications and hardware components corresponding to the collection of components, wherein the plurality of software applications and hardware components meets the plurality of application requirements, and wherein specifying the underlying infrastructure further comprises: estimating a performance of the underlying infrastructure according to the system design by aggregating the user inputted estimated performance of each of the components in the composite application system design, wherein the estimated performance is evaluated by comparing the estimated performance with system performance requirements, which are user-specified by thresholds and computations and if the estimated performance is larger than or equal to the system performance requirements, then the estimated cost of the composite application system design is

further compared with system budget requirements; and identifying potential flaws of feasible designs by determining the specific components that cause the system design to fail to meet system requirements by comparing the estimated performance and estimated cost of each component in the composite application system design with an expected performance and expected cost of a component that meets system requirements, further wherein the process is performed by an integrated software application, wherein the composite application system design is implemented as one or more of a plurality of executables, software programs, software packages, data structures, file systems and configuration data.

[0012c] In another aspect, there is provided a system for implementing a collaborative design application for simulating the performance of a composite application system according to a system objective, the collaborative design application comprising: an interface for receiving an input defining a plurality of case scenarios performed by a composite application system in a system design and for receiving an input specifying an underlying infrastructure for the system design to embody the plurality of components, wherein specifying the underlying infrastructure includes specifically selecting a plurality of software applications and hardware components corresponding to the collection of components, wherein the plurality of software applications and hardware components meets the plurality of application requirements; a repository, communicatively coupled to the interface, for storing completed composite application system designs; a database, communicatively coupled to the interface, for collecting data of individual information technology objects according to a plurality of metrics; a first viewer, coupled to the interface, for displaying

volumetric load data and performance metric data of the composite application system design; and a second viewer, coupled to the interface, for graphically representing relationships between information technology objects in the composite application system design, wherein the system is implemented as a computer-readable medium storing a computer-executable integrated software application, the integrated software application when executed by at least one processor causing the at least one processor to: simulate the performance of said composite application system comprising a plurality of components comprising a collection of software applications and hardware components, and integrate the plurality of case scenarios with a plurality of components of the composite application system, and wherein specifying the underlying infrastructure further comprises: estimating a performance of the underlying infrastructure according to the system design by aggregating the user inputted estimated performance of each of the components in the composite application system design, wherein the estimated performance is evaluated by comparing the estimated performance with system performance requirements, which are user-specified by thresholds and computations and if the estimated performance is larger than or equal to the system performance requirements, then the estimated cost of the composite application system design is further compared with system budget requirements; identify potential flaws of feasible designs by determining the specific components which cause the system design to fail to meet system requirements by comparing the estimated performance and estimated cost of each component in the composite application system design with an expected performance and expected cost of a component that meets system requirements wherein further, the design of the

composite application system comprises the input of a plurality of disparate contributors received from the interface.

BRIEF DESCRIPTION OF DRAWINGS

[0013] The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

[0014] Figure 1 depicts a component diagram of an integrated design application for designing composite application systems, in accordance with one embodiment.

[0015] Figure 2 depicts a flowchart of a process of designing a composite application system, in accordance with one embodiment.

[0016] Figure 3 depicts a flowchart of a process of specifying an underlying infrastructure for a system design, in accordance with one embodiment.

[0017] Figure 4 is a block diagram of an exemplary computer system upon which embodiments of the present invention may be implemented, in accordance with one embodiment.

[0018] Figure 5 depicts a graphical representation of an exemplary user interface for defining use case scenarios during a process of designing a

composite application in an integrated design application system, in accordance with one embodiment.

[0019] Figure 6 depicts graphical representation of an exemplary user interface for integrating use cases with execution tools during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0020] Figure 7 depicts a graphical representation of an exemplary user interface for specifying an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0021] Figure 8 depicts a graphical representation of a user interface for mapping relationships between technical processes in a use case scenario with execution tools during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0022] Figure 9 depicts a graphical representation of a user interface for mapping relationships between components in a use case scenario with components of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0023] Figure 10A depicts a graphical representation of a user interface for defining volumetric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment

[0024] Figure 10B depicts a graphical representation of a user interface for defining target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment

[0025] Figure 11 depicts a graphical representation of a user interface for applying volumetric and target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment

[0026] Figure 12 depicts an alternate graphical representation of a user interface for applying volumetric and target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0027] Figure 13 depicts a graphical representation of an exemplary performance file corresponding to a selected object in an underlying

infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0028] Figure 14 depicts a graphical representation of an exemplary comparison interface corresponding to volumetric load data and performance metric data of a selected object in an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0029] Figure 15 depicts a graphical representation of an exemplary Metric and Volumetric Display corresponding to a selected object in a layout of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

DETAILED DESCRIPTION

[0030] Reference will now be made in detail to several embodiments. While the subject matter will be described in conjunction with the alternative embodiments, it will be understood that they are not intended to limit the claimed subject matter to these embodiments. On the contrary, the claimed subject matter is intended to cover alternative, modifications, and equivalents, which may be included within the spirit and scope of the claimed subject matter as defined by the appended claims.

[0031] Furthermore, in the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the claimed subject matter. However, it will be recognized by one skilled in the art that embodiments may be practiced without these specific details or with equivalents thereof. In other instances, well-known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects and features of the subject matter.

[0032] Portions of the detailed description that follows are presented and discussed in terms of a method. Although steps and sequencing thereof are disclosed in a figure herein (e.g., Figures 2 and 3) describing the operations of this method, such steps and sequencing are exemplary. Embodiments are well suited to performing various other steps or variations of the steps recited in the

flowchart of the figure herein, and in a sequence other than that depicted and described herein.

[0033] Some portions of the detailed description are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits that can be performed on computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer-executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0034] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout, discussions utilizing terms such as “accessing,” “writing,” “including,”

"storing," "transmitting," "traversing," "associating," "identifying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0035] In the following embodiments, an approach is described for providing an integrated design application for designing composite applications. Embodiments propose a method and system for designing composite applications that provides users with multiple views of the processes included in the design, allows users to explicitly create and view relationships across these views, allows assigning volumetric load information and target metrics to technical processes and to choose the infrastructure components whose performance match or approximate the target metrics for a specified load.

INTEGRATED DESIGN APPLICATION

[0036] With reference now to Figure 1 a component diagram of an integrated design application 100 for designing composite application systems is depicted, in accordance with one embodiment. In a typical configuration, integrated design application 100 includes a client application 101, a business logic component 103 and a backend storage component 105.

[0037] According to one embodiment, a user 111 of the integrated design application 100 accesses the functionality of the integrated design application 100 via client device 113. Client device 113 may be the environment upon which interface 115 of the integrated design application 100 is implemented and accessed by the user 111. In some embodiments, a plurality of distributed users may utilize a plurality of client devices to access functionality of integrated design application 100 through a plurality of locally installed copies of interface 115.

[0038] In one embodiment, interface 115 of integrated design application 100 allows user 111 to manipulate (e.g., create, capture, provision, account for, etc...) composite application system designs. Interface 115 may, for example, allow user 111 to design a composite application system by provisioning a plurality of components having various attributes performing one or more case scenarios, to represent relationships among the components, and to simulate the performance of the designed system and specific components according to various metrics. Interface 115 may also be implemented to allow user 111 to view and make modifications to the design (i.e., by provisioning one or more substitute components) to more precisely align with system requirements or objectives.

[0039] Interface 115 may also capture (i.e., comprehensively represent) a composite application system design across a plurality of design "layers." For

example, a design, or portions of a design, may be represented across a process layer (e.g., the functions performed by the system), a logic layer (e.g., the information and/or technology tools required to perform the functions) and a physical layer (e.g., the actual, specifically identified devices selected to be used as the tools to perform the functions). In alternative embodiments, the number of layers comprising a composite application system design is not limited to the three specifically enumerated layers provided above (e.g., other layers may include a virtualization layer to show virtualized infrastructure and an application layer to show interactions of the software applications involved). According to such embodiments, a user 111 may allocate and specify any number of layers to represent the design.

[0040] Whereas conventional techniques for designing a composite application system may include designing the system over a plurality of design layers by using one or more independent, distinct platforms and design applications to represent each layer separately, integrated design application 100 allows a user 111 to view each layer in the design both individually and as the combination of the plurality of layers within a single application and interface 115. Furthermore, each of the plurality of layers comprising a composite application system design in the integrated design application 100 is capable of user-defined interrelation, thus cross referencing of the components (e.g., tools, relationships and case scenarios) is possible through the integrated design application 100. Accordingly, the aggregate effect on the system for each component may be

visually represented to the user 111 through the interface 115 of the integrated design application 100.

[0041] Interface 115 may also include functionality to correlate material from disparate contributors to a project, to leverage past work as templates, to assign and display metric data, or to package and transfer designs (or files comprising specific portions of a design) to recipients. Correlating material from disparate contributors to a project may include, for example, merging functionality between nonconforming versions of the same file or file(s). Conflicting items discovered during the merging process may be displayed to the merging user via interface 115. Within these embodiments, the user may be able to dictate the result of merging two conflicting versions of the same item.

[0042] Interface 115 may also correlate material from disparate contributors to a project by offering communication functionality, such as messaging. For example, a user 111 may be able to attach one or more notes (or other textual/graphical displays) to a file, or a specific object in a file. Additional correlative functionality includes allowing documentation features for collaborating users. In some embodiments, a user may also be able to export (i.e., duplicate and output) design documentation in word processor and other text inclusive file types (e.g., as documents), spreadsheets, or graphic diagramming applications. Interface 115 may leverage past work as templates by, for example, accessing and displaying stored existing architecture (e.g.,

designs or portions of designs) stored in local storage, or backend storage component 105.

[0043] Typically, the design is stored locally in the client application as one or more files, collectively referred to and/or distinguished as a project. For projects which include contributions from distributed collaborators, a user 111 is able to transfer the locally stored version of the design (or portions of the design) to other users. In some embodiments, the user is able to transfer the design to a central repository accessible to other collaborators. In other embodiments, client devices of collaborating users also operate as nodes in a peer-to-peer network. In such embodiments, the transfer of files may be performed directly between client devices, in lieu of transferring to a central repository. Interface 115 may allow the user 111 to initiate connection with other collaborators, and may allow user-selection of source and destination of files. Transfer of the design may be performed in whole - typically by packaging the files comprising the project - or in part (transferring one or more specific files).

[0044] In some embodiments, interface 115 is implemented to include embedded applications that provide the underlying functionality accessible to the user 111. Embedded applications may, for example, include a diagramming application interface, word processing application, spreadsheet application, local database for local work storage, and one or more viewing applications for graphically representing relationships between components in the system. In

alternate embodiments, the diagramming application interface may itself include functionality similar to word processing applications and/or may include features that graphically represent relationships between components in the system.

[0045] According to some embodiments, a design for a composite application system may include certain parameters according to specified input load and output metrics related to the system. Typical load information is characterized by volumetric data. This volumetric data may include, for example, the number/size of user sessions, number/size of messages, frequency and duration by time (e.g., hour, day, year, etc.) of access. Typical output metrics relate to system performance, and/or cost. Typically, load metrics have direct and/or indirect correlation with output metrics. Loads and metrics may be pre-defined, or may be customizable by the user, according to various embodiments. Interface 115 may be utilized to enter metrics according to user specified data. In some embodiments, load data associated with use case or application components and/or metric data corresponding to specific identified physical components may be supplied from external sources (e.g., backend storage component 105). In alternate embodiments, load and metric data may be customized and stored locally, for example, via programming in an embedded spreadsheet application.

[0046] Business logic component 103 facilitates collaboration among one or more distributed users of integrated design application 100. Facilitating

collaboration may include, for example, rationalizing versions of a composite system design locally stored in the client devices of a plurality of distributed users with a central ("master") version. Business logic component 103 may provide additional collaboration-centric services such as providing advanced design simulation, granting access to external data sources, and capturing volumetric load data, among others. Business logic component 103 may include, in one embodiment, one or more computing devices (e.g., servers 121) communicatively coupled to each other, one or more client devices, or a network accessible by one or more client devices.

[0047] In one embodiment, the transfer of files between distributed users of design projects in integrated design application 100 is performed through business logic component 103. A user seeking to transfer a project or a subset of files to another collaborator (or to update a central version) sends the project (or subset of files) to business logic component 103. Business logic component 103 may subsequently archive and version these projects. Subsequent authorized users may be granted access to one or more versions of the project (or subsets of files).

[0048] In further embodiments, business logic component 103 includes advanced version control features. For example, business logic component 103 may execute features such as automatic documentation, exclusive checkout, version merging, version snapshotting and version rollback. Automatic

documentation updates the metadata corresponding to usage of the file (e.g., dates and times of creation, last modification, etc...) and includes automated updating of the content therein. Exclusive checkout restricts access to one or more versions of a design project. For example, where checkout is restricted to one collaborator, once one or more versions of the design project stored in the business logic component 103 are accessed by a user 111, no other user is able to access the one or more versions of the design project until the one or more versions of the design project are no longer being accessed (i.e., resubmitted).

[0049] In some embodiments, exclusive checkout may be restricted to the version (i.e., other users may access versions not currently being accessed). In alternate embodiments, the length of time a checkout maintains exclusivity is limited to a period of duration. The expiration of said period removes the restrictions on access to the files being checked out.

[0050] Version merging reconciles multiple changes made to different copies of the same file. These changes are merged, resulting in a single new file that contains both sets of changes. In some embodiments, an authorized user is able to utilize interface 115 to dictate the result of merging two conflicting versions of the same item or file. The merge may be executed within business logic component 103, prior to transmission to interface 115 of one or more users. The merging user may also update the version of the file stored in the central repository with the merged version in business logic component 103.

[0051] Version snapshotting preserves versions of files which have been updated or merged. According to one embodiment, files within the central repository which have been updated may have previous versions renamed, stored, and may remain accessible to collaborating users. Likewise, version rollback allows an authorized user with access to the project to return a current version of a file to a previous version. For example, the current (i.e., most recently updated) master version of a file may be reconverted to a previous version, which becomes the master version.

[0052] For each of the features described (i.e., automated system documentation, exclusive checkout, version merging, version snapshotting and version rollback), authorized users may include, for example, all collaborators in a project, or be limited to specific users within a project. In other embodiments, authorization may be granted for the contents of the project. In another embodiment, authorization may be specifically limited to one or more files. Exclusive checkout, version merging, version snapshotting and version rollback are all features well known in the art, with various known implementations. Accordingly, a detailed explanation of implementation shall be omitted herein.

[0053] Version control data may also be explicitly captured by business logic component 103. Version control data may include, for example, the number of collaborators on a project, the number of transactions (i.e., file transmissions)

occurring during the duration of a project design, the size of communication (e.g., messages), and chronological details (such as time of day, time of year) for transactions and/or communications performed during the project life. Some version control data may be obtained, for example, by monitoring the number of identities (e.g., IP addresses, or user accounts in a network) accessing the central versions of the files. Alternatively, some volumetric data may also be obtained by monitoring the number of modifications (e.g., updates) to central versions of the files.

[0054] In yet further embodiments, business logic component 103 may extend features such as predictive operations modeling, run-time optimizing, and real time automated provisioning.

[0055] Backend storage component 105 provides additional storage functionality. Backend storage component 105 may be implemented as a repository storing versioning for users of integrated design application 100. In some embodiments, backend storage component 105 includes a database 131 of component attributes. For example, database 131 may store a knowledge base of vendors and the products corresponding to each vendor, with known attributes such as cost and performance. Attributes may include vendor-provided information. In further embodiments, backend storage component 105 collects performance metrics in configured systems for leverage in predicting

performance of future systems. Backend storage component 105 may also store existing architectures which may be referenced by users for use as templates.

[0056] By providing the ability to correlate material from multiple (remote) contributors, the integrated design application 100 provides the ability for multiple users 111 to collaborate on the same project. Moreover, collaboration is possible directly amongst contributors, and during the same stage or phase of the design. Whereas conventional techniques which incorporate collaboration may exclude simultaneous and concurrent modifications and/or require manual conversion between disparate contributors, the above described features provide additional functionality to facilitate a seamless collaboration and subsequent content conversion.

COMPOSITE APPLICATION SYSTEM DESIGN

[0057] Figure 2 depicts a flowchart 200 of a process of designing a composite application system in an integrated design application, in accordance with one embodiment. Steps 201-205 describe exemplary steps comprising the process 200 in accordance with the various embodiments herein described.

[0058] At step 201, one or more use case scenarios of an ideated solution for an application system design are defined. A use case describes a sequence of behaviors in a system. Each use case describes how an actor (i.e., person or object supplying a stimulus to the system) interacts with the system to achieve a

specific goal. One or more scenarios may be generated from each use case, with these scenarios corresponding to the available methods of achieving the goal.

[0059] Use case scenarios are often characterized as either a business use case or a system use case. A business use case generally describes a business process which provides value to a business actor, is typically described without high-level technical terminology, and describes what the process does. A system use case is normally described at the system functionality level and specifies the function or service the system provides for the user. A system use case will describe what the actor achieves through interaction with the system.

[0060] Defining use case scenarios for an ideated solution may include defining the scenarios that depict the functionality of the system with respect to the stages, programs, operations, processes and interactions reached or performed by the system to execute the functions. A use case may, for example, comprise the "steps" performed by a system to execute a specific operation, or to achieve an intended result. In addition, volumetric load and outcome target metrics may be associated with the use case.

[0061] At step 203, the one or more use case scenarios defined in step 201 are integrated with appropriate information technology tools for executing the use case scenarios defined in step 201. These information technology tools

consist of the technology (e.g., components, devices) and applications (software) necessary to perform the use case scenarios. For example, a simple web-based business may include providing a service through one or more software platforms executing on one or more storage/processing components (e.g., servers) coupled to one or more communication technology components (e.g., routers).

[0062] At step 205, the underlying infrastructure comprising the composite application system is specified. Specifying the underlying infrastructure comprising the composite application system may include identifying the specific hardware and software components embodying the one or more execution tools. Specifying the underlying infrastructure typically also includes specifying the combination of hardware and software components that achieves system objectives optimally (i.e., with the greatest efficiency and/or efficacy). Specifying an underlying infrastructure may also include identifying the combination of hardware and software components which best approximates achievement of system objections, where actual achievement is impractical.

[0063] Whereas the system according to Figure 2 has been designed according to three model layers, it is to be understood that a system design is not limited to the three exemplary model layers. Rather, the number of model layers may vary according to both usage and embodiments.

SPECIFYING UNDERLYING INFRASTRUCTURE

[0064] Figure 3 depicts a flowchart 300 of a method of specifying an underlying infrastructure for a composite application system design as discussed in step 205, in accordance with one embodiment.

[0065] At step 301, the performance of the underlying infrastructure of the composite application system designed is estimated. According to one embodiment, estimating the performance of a system design may be performed by aggregating the user inputted estimated performance of each of the components in the component application system design. In some embodiments, the performance data of the components comprising the component application system design is pre-stored and accessible within the integrated design application.

[0066] At step 303, the feasibility of the design according to project objectives and/or goals is evaluated. Evaluation of the feasibility of the design comprises evaluating the estimated performance of the composite application system design obtained in step 301. Evaluation may comprise comparing the estimated performance of a composite application system according to the design with system performance requirements. In one embodiment, users may specify the thresholds and computations used to estimate feasibility of composite applications via an associated programmable platform (e.g., a spreadsheet). A successful comparison (i.e., wherein the estimated performance of the composite application system according to the design meets or exceeds system

performance requirements) may be further evaluated to compare the estimated cost of the composite application system design with the system budget requirements.

[0067] A system design which meets both system budget and performance requirements is considered feasible. Alternatively, composite application system designs which fail either the performance evaluation or the cost evaluation by a relatively small margin may also be considered feasible. Composite application system designs which fail both performance and cost evaluations, or which fail either performance or cost evaluations by a proportionally large margin may be evaluated as unfeasible.

[0068] At step 305, potential flaws of feasible designs (according to the evaluation performed at step 303) are identified. Potential flaws may include the specific component or components which cause the system design to fail to meet system requirements. Potential flaws may also include a specific combination or arrangement of components which result in suboptimal output. Determination of the specific components which cause the system design to fail may be made by comparing the estimated performance and estimated cost of each component in the composite application system design with the expected performance and expected cost of a component that meets system requirements.

BASIC COMPUTING DEVICE

[0069] Referring now to Figure 4, a block diagram of an exemplary computer system 412 is shown. It is appreciated that computer system 412 described herein illustrates an exemplary configuration of an operational platform upon which embodiments may be implemented to advantage. Nevertheless, other computer systems with differing configurations can also be used in place of computer system 412 within the scope of the present invention. That is, computer system 412 can include elements other than those described in conjunction with Figure 4. Moreover, embodiments may be practiced on any system which can be configured to enable it, not just computer systems like computer system 412. It is understood that embodiments can be practiced on many different types of computer system 412. Examples include, but are not limited to, desktop computers, workstations, servers, media servers, laptops, gaming consoles, digital televisions, PVRs, and personal digital assistants (PDAs), as well as other electronic devices with computing and data storage capabilities, such as wireless telephones, media center computers, digital video recorders, digital cameras, and digital audio playback or recording devices.

[0070] With reference to Figure 4, an exemplary system for implementing embodiments includes a general purpose computing system environment, such as computing system 412. In its most basic configuration, computing system 412 typically includes at least one processing unit 401 and memory, and an address/data bus 400 (or other interface) for communicating information. Depending on the exact configuration and type of computing system

environment, memory may be volatile (such as RAM 402), non-volatile (such as ROM 403, flash memory, etc.) or some combination of the two. Computer system 412 may also comprise an optional graphics subsystem 405 for presenting information to the computer user, e.g., by displaying information on an attached display device 410, connected by a video cable 411.

[0071] Additionally, computing system 412 may also have additional features/functionality. For example, computing system 412 may also include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Such additional storage is illustrated in Figure 4 by data storage device 404. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. RAM 402, ROM 403, and data storage device 404 are all examples of computer storage media.

[0072] Computer system 412 also comprises an optional alphanumeric input device 406, an optional cursor control or directing device 407, and one or more signal communication interfaces (input/output devices, e.g., a network interface card) 408. The optional alphanumeric input device 406 can communicate information and command selections to central processor 404. The optional cursor control or directing device 407 is coupled to bus 400 for communicating user input information and command selections to central

processor 404. The signal communication interface (input/output device) 408, which is also coupled to bus 400, can be a serial port. The communication interface 408 may also include wireless communication mechanisms. Using communication interface 408, computer system 412 can be communicatively coupled to other computer systems over a communication network such as the Internet or an intranet (e.g., a local area network), or can receive data (e.g., a digital television signal).

DEFINING USE CASE SCENARIOS

[0073] Figure 5 depicts a graphical representation of an exemplary user interface 500 for defining use case scenarios during a process of designing a composite application in an integrated design application system (step 201, Fig. 2), in accordance with one embodiment.

[0074] In the depicted embodiment, user interface 500 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 500 includes toolbar 501, panel 503, viewing window 505, exemplary design layout (e.g., layout window 507), File Browser 511, File Information Display 513, Object Information Display 515, Object Target Window 509 and Metric and Volumetric Display 517.

[0075] Toolbar 501 provides access to functions available to the user 111 of the integrated design application. Toolbar 501 typically is implemented to

display keywords approximating a general characterization of the underlying functions. Often, one or more functions are further revealed to the user via a drop down bar or menu by clicking (e.g., selecting via an input device) the keyword. As shown, toolbar 501 includes functions related to: manipulating the file ("File"), diagramming tools ("Visio Tools"), calculation (e.g., "Compute"), relationship mapping ("Relationships"), target output metric and load volumetric data ("Performance") and communication ("Messaging"). Accordingly, other embodiments may include different functions, a different arrangement of the enumerated functions, or may include a subset of the one or more functions.

[0076] As presented, panel 503 displays a panel of operation-specific functions. For example, in the current depiction of Figure 5, user interface 500 displays the process layer of an exemplary design. As such, panel 503 includes functions specific to the current operation (in the present case, defining use case scenarios). As shown, panel 503 includes "Shapes" comprising shaped objects representative of components which may be used in defining use case scenarios.

[0077] As shown, viewing window 505 displays the current file or content the user is viewing and/or manipulating. As depicted in Figure 5, viewing window 505 includes a display of layout window 507, which displays the use case(s) comprising the exemplary project. In further embodiments, viewing window may include support for the content displayed. For example, during design or display

operations, features that alter perspective (e.g., “Pan & Zoom”) or simple tools that assist in measurement may be provided.

[0078] As depicted, layout window 507 depicts an exemplary use case scenario depicting the technical processes included in exemplary design. The objects comprising layout window 507 are exemplary and for display purposes only.

[0079] File Browser 511 (“Project File Browser”) displays the files currently collected in the project. As depicted, File Browser 511 further distinguishes the files contained in the project according to layer. In other embodiments, File Browser 511 may automatically organize the files contained in File Browser 511 according to other arrangements. In further embodiments, File Browser 511 may display the files in an order specified by a collaborating user. Alternatively, the order of files displayed may be specific to the user, and/or adjustable for each version of the design.

[0080] In one embodiment, File Information Display 513 may display information corresponding to the currently viewed file(s). For example, File Information Display 513 may indicate the existence of relationships of objects within the currently viewed file(s). In other embodiments, File Information Display 513 may also provide alerts and/or notice that one or more objects in a file

conflict with system objectives. In further embodiments, File Information Display 513 may display collaborator notes (e.g., messages to other collaborators).

[0081] As presented, Object Information Display 515 displays information corresponding to a selected object or process. The information displayed in Object Information Display 515 may, for example, include computational characteristics of the selected object, and/or metadata corresponding to estimated performance statistics. Computational characteristics may include computations for the data entered in Object Information Window 509 corresponding to the selected object or process. Metadata corresponding to estimated performance statistics may, for example, include the number of reporting data sources comprising the estimation. According to some embodiments, when no object or process is currently selected, Object Information Display 515 may display the information of the last selected object or process. In further embodiments, when no object or process is selected, Object Information Display 515 may be empty. In alternate embodiments, when no object or process is selected, Object Information Display 515 may display an indication that no object or process is selected.

[0082] As depicted, Object Target Window 509 displays information corresponding to a selected object or process in a use case, as referenced by Object Information Display 515. Information corresponding to a selected object or process in a use case may include target metrics (i.e., the conditions sufficient

to achieve system objectives) for that object or process. In some embodiments, Object Target Window 509 allows a viewing user with authorization to enter target data for the selected object or process. For exemplary purposes, as presented, Object Target Window 509 allows a user to view and/or set a target cost, availability and delay for the selected object.

[0083] Metric and Volumetric Display 517 displays information corresponding to metrics established for a selected object or process. The information contained in Metric and Volumetric Display 517 may be used, for example, to view the performance of the system design relative to system objectives and/or targeted metrics (e.g., cost, performance, etc...). A user is thus able to view whether the metrics of the system design conform with the objectives or targeted metrics. Accordingly, nonconforming system designs may be modified (e.g., re-provisioning components of the underlying infrastructure) to more precisely align with the system. Information corresponding to metrics established for a selected object or process may include data fields such as designated metric goals, estimated performance of current design, absolute performance limits according to components comprising current design, etc... Information corresponding to metrics established for a selected object or process may be obtained from a pre-established storage of data and/or may be entered by a user via Object Information Window 509.

[0084] In further embodiments, metric and volumetric display 517 may provide the calculated estimates for metric data according to the current system design. Metric data may include, for example, cost, carbon, performance and availability metrics. Metric data may be obtained via referenced to a depository of metric data corresponding to the components comprising the current system design. Modifications to the current system design may thus result in a running recalculation and modified estimates for the metric data of the system design. Accordingly, metric data estimates may be available during the design process, and a completed system design would not be necessary to derive estimated metric data according to this embodiment. In still further embodiments, the metric and volumetric display 517 re-calculates composite metrics for the current system design across all application layers on demand.

[0085] By providing the ability to display calculated estimates for metric data, a user is able to immediately monitor the effect to the composite system design of each additional component or relationship added. Subsequently, system goals may be easier to track, and system feasibility may be determined earlier during the design process than according to conventional techniques. For example, over provisioning may be detected immediately if the combination of certain components has an estimated cost greater than according to the application or project requirements.

[0086] According to some embodiments, when no object or process is currently selected, Metric and Volumetric Display 517 may display the information of the last selected object or process. In further embodiments, when no object or process is selected, Object Information Display 515 may be empty. In alternate embodiments, when no object or process is selected, Object Information Display 515 may display an indication that no object or process has been selected. In yet further embodiments, Metric and Volumetric Display 517 may explicitly indicate that an object or process lacking sufficient volumetric load data and/or performance metric data to compute a particular data field has insufficient data.

USE CASE AND EXECUTION TOOL INTEGRATION

[0087] Figure 6 depicts a graphical representation of an exemplary user interface 600 for integrating use cases with execution tools during a process of designing a composite application in an integrated design application system (step 203, Fig. 2), in accordance with one embodiment.

[0088] In the depicted embodiment, exemplary user interface 600 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 600 includes toolbar 501, panel 503, viewing window 505, exemplary design layout (e.g., layout window 607), File Browser 511, File Information Display 513, Object Information Display 515, Object Information Window 509, Metric and Volumetric Display 517, and Related Technical Processes display 619.

[0089] Detailed descriptions for toolbar 501, panel 503, viewing window 505, File Browser 511, File Information Display 513, Object Information Display 515, Object Information Window 509, and Metric and Volumetric Display 517 have been provided have been provided in the foregoing embodiments. Accordingly, repetition shall be herein omitted.

[0090] As depicted, layout window 607 depicts an exemplary use case scenario having components which are integrated with execution tools in an exemplary composite application design. The objects comprising layout window 607 are exemplary and for display purposes only. As shown, layout window 607 includes execution tools which have been paired to specific technical processes in an exemplary use case scenario. For example, one technical process in the use case scenario is displayed as "Patrol Monitoring Identifies an App Server which has failed." Layout window 607 displays the execution tool (e.g., "Marimba Agent") which performs the action identified by the technical process.

[0091] As presented, Related Technical Processes display 619 displays the related technical processes for a selected object (e.g., execution tool, infrastructure device) in layout window 607. Related technical processes may include the technical processes mapped to the object. Technical processes may include the operations performed by the object to enable achievement of the specific use defined in the use case. Objects which have no related technical

processes may have an empty Related Technical Processes display 619. In alternative embodiments, objects which have no related technical processes may have a Related Technical Processes display 619 explicitly indicating the absence of related technical processes.

SPECIFYING UNDERLYING INFRASTRUCTURE

[0092] Figure 7 depicts a graphical representation of an exemplary user interface 700 for specifying an underlying infrastructure during a process of designing a composite application in an integrated design application system (step 205, Fig. 2), in accordance with one embodiment.

[0093] In the depicted embodiment, exemplary user interface 700 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 700 includes toolbar 501, panel 503, viewing window 505, exemplary design layout (e.g., layout window 707), File Browser 511, File Information Display 513, Object Information Display 515, Object Information Window 509, Metric and Volumetric Display 517, and Related Technical Processes display 619.

[0094] Detailed descriptions for toolbar 501, panel 503, viewing window 505, File Browser 511, File Information Display 513, Object Information Display 515, Object Information Window 509, Metric and Volumetric Display 517, and

Related Technical Processes display 619 have been provided in the foregoing embodiments.

[0095] As depicted, layout window 707 depicts the components comprising an underlying infrastructure in an exemplary composite application design. The objects comprising layout window 807 are exemplary and for display purposes only. As shown, layout window 707 includes specifically identified hardware devices and software platforms which have been selected as the tools to execute the use cases of a use case scenario. For example, objects designated "virtualization layer-Vmware ESX" each reference a VMware ESX Server - an enterprise-level virtualization product offered by commercial vendor VMware, Inc.

RELATIONSHIP MAPPING BETWEEN USE CASE COMPONENTS AND EXECUTION TOOLS

[0096] Figure 8 depicts a graphical representation of a user interface 800 for mapping relationships between technical processes in a use case scenario with execution tools during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[0097] In the depicted embodiment, user interface 800 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 800 includes one or more viewing surfaces (e.g., viewing windows 801, 805) and exemplary design layouts (e.g., layout windows

803, 807). Exemplary user interface 800 also includes Relevant Elements Display 809, and Ordered Relationship Display 811.

[0098] Viewing windows 801 and 805 display layout windows (e.g., layout windows 803 and 807) and allow a user to view and/or map relationships between components (e.g., technical processes and execution tools) comprising the layout windows. For example, layout window 803 displays the one or more technical processes comprising a use case scenario for an exemplary composite application design. Layout window 807 displays the use case scenario with execution tools. User interface 800 thus allows a user to integrate the technical processes of the use case scenario displayed in layout window 803 with the execution tools in layout window 807 by mapping a relationship between a technical process and its corresponding execution tool. Accordingly, once mapped, the relationship is explicitly observable through user interface 800.

[0099] Ordered Relationships Display 811 displays a list of the relationships between objects in layout windows 803 and 807 of user interface 800 which have been mapped. Ordered Relationships Display 811 may also allow a user to create (i.e., "map") and delete (i.e., "un-map") relationships between objects in the layout windows. In some embodiments, Ordered Relationships Display 811 may also allow a user to save modifications to existing mapped relationships. In some embodiments, Ordered Relationships Display

811 also allows a user to alter the presentation of the list of mapped relationships in Ordered Relationships Display 811.

[00100] According to further embodiments, when no relationships have been mapped, Ordered Relationships Display 711 may be empty. In alternate embodiments, when no relationships have been mapped, Ordered Relationships Display 811 may display an indication that relationships have been mapped.

RELATIONSHIP MAPPING BETWEEN USE CASE COMPONENTS AND UNDERLYING INFRASTRUCTURE

[00101] Figure 9 depicts a graphical representation of a user interface 900 for mapping relationships between components in a use case scenario with components of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00102] In the depicted embodiment, user interface 900 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 900 includes one or more viewing surfaces (e.g., viewing windows 801, 905) and exemplary design layouts (e.g., layout windows 803, 907). Exemplary user interface 900 also includes Relevant Elements Display 809, and Ordered Relationship Display 811.

[00103] Viewing windows 801 and 905 display layout windows (e.g., layout windows 803 and 907) and allow a user to view and/or map relationships between components (e.g., technical processes and underlying infrastructure devices) comprising the layout windows. For example, layout window 803 displays the one or more technical processes comprising a use case scenario for an exemplary composite application design. Layout window 907 displays the underlying infrastructure of the same exemplary composite application design. User interface 900 thus allows a user to map a relationship between a technical process displayed in layout window 803 with a specific device comprising the infrastructure of the use case scenario displayed in layout window 907. Accordingly, once mapped, the relationship is explicitly observable through user interface 800.

[00104] Detailed descriptions for Relevant Elements Display 809 and Ordered Relationships Display 811 have been provided have been provided in the foregoing embodiments.

DEFINING VOLUMETRIC AND TARGET METRIC DATA

[00105] Figures 10A and 10B depict graphical representations of a user interface 1000, 1100 for, respectively, defining volumetric and target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00106] In the depicted embodiment, user interface 1000 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 1000 includes one or more data fields (e.g., title field 1001 and unit abbreviation field 1003) and action buttons (e.g., cancel button 1005 and finish button 1007).

[00107] User interface 1000 may, for example, be instantiated by user interface 500 via navigation through toolbar 501. User interface 1000 allows the definition of volumetric data according to user input. As depicted in Figure 10A, the user is able to define a volumetric by specifying a name (via title field 1001) and the abbreviation of units corresponding to the volumetric (via unit abbreviation field 1003). Definition of a volumetric may include alternate and/or additional fields according to usage and embodiments.

[00108] After entering data defining the volumetric, the user is able to submit the input via finish button 1007. After submission, the volumetric is added to a corresponding storage component of the integrated design application system, and the volumetric is stored and persists (e.g., may be used repeatedly) for the current project (e.g., during the design of the present composite application). In other embodiments, a user may elect to store the volumetric permanently (until manually deleted) in a storage component of the integrated design application system, such that the volumetric may be used in subsequent

composite application designs. Cancel button 1005 allows the user to exit interface 1000 without submitting data entered in the data fields.

[00109] In the depicted embodiment, user interface 1100 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 1100 includes one or more data fields (e.g., title field 1101 and unit abbreviation field 1103) and action buttons (e.g., cancel button 1105 and finish button 1107).

[00110] User interface 1100 may, for example, be instantiated by user interface 500 via navigation through toolbar 501. User interface 1100 allows the definition of performance metric data according to user input. As depicted in Figure 10B, the user is able to define a volumetric by specifying a name (via title field 1001) and the abbreviation of units corresponding to the performance metric (via unit abbreviation field 1003). Definition of a performance metric may include alternate and/or additional fields according to usage and embodiments.

[00111] After entering data defining the performance metric, the user is able to submit the input via finish button 1007. After submission, the performance metric is added to a corresponding storage component of the integrated design application system, and the performance metric is stored and persists (e.g., may be used repeatedly) for the current project (e.g., during the design of the present composite application). In other embodiments, a user may elect to store the

performance metric permanently (until manually deleted) in a storage component of the integrated design application system, such that the performance metric may be used in subsequent composite application designs. Cancel button 1005 allows the user to exit interface 1000 without submitting data entered in the data fields.

APPLYING VOLUMETRIC AND TARGET METRIC DATA TO OBJECT

[00112] Figure 11 depicts a graphical representation of a user interface 1200 for applying volumetric and target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00113] In the depicted embodiment, user interface 1200 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 1200 includes metric selection field 1201, metric definition field 1203 and action button (e.g., finish button 1007).

[00114] User interface 1200 may, for example, be instantiated by user interface 500 via navigation through toolbar 501. User interface 1200 allows the application of volumetric and performance data to a selected object in a composite application design. As depicted in Figure 11, the user is able to apply a volumetric or performance metric by selecting the name of a pre-defined volumetric or performance metric in metric selection field 1201. In one

embodiment, metric selection field 1201 displays all volumetrics and performance metrics defined for the selected object in the composite application design. In other embodiments, metric selection field 1201 may also display volumetrics and performance metrics for all related objects (e.g., objects identified as descendants, ancestors or siblings of the selected object), and/or for an entire group of objects of which the selected object belongs.

[00115] After selecting a volumetric or performance metric for the selected object, the data comprising the selected volumetric (e.g., boundaries) may be input by the user via metric definition field 1203. As presented in Figure 11, a volumetric selected in metric selection field 1201 (e.g., "Peak user transactions") may comprise sub-fields including timescales (the scale of time used in a calculation or measurement, e.g., "Hourly," "Daily," "Weekly," etc. and boundaries (the duration or range of time and dates to measure or calculate)

[00116] After data comprising the selected volumetric is entered via metric definition field 1203, a user may apply the volumetric or performance metric to the selected object by selecting finish button 1207. In one embodiment, after a volumetric or performance metric has been applied to an object, a corresponding performance file is edited for the selected object to include the newly applied volumetric or performance metric. The performance file may, for example, be implemented as a spreadsheet comparing the performance data entered for the object with the volumetric load data entered for the object.

[00117] In other embodiments, after a volumetric or performance metric has been applied to the selected object, the volumetric or performance metric data is added to every instance of the object in the composite application design. In further embodiments, a user may elect to apply the volumetric or performance data to all objects of same type of the selected object permanently (until manually deleted) in a storage component of the integrated design application system, such that all subsequent instances of the selected object will include the volumetric or performance data.

[00118] In an alternate embodiment, the user may select to use baseline (e.g., default) volumetric and metric definitions and quantities from a data repository which correspond to the project type. For example, the user may be able to select from a menu and choose a pre-defined item from which to model the project's definitions based on a pre-existing set of data that is stored and available from the central repository. The application then makes available those typical volumetric types and associated metrics corresponding to each volumetric type, and may automatically pre-fill them in the spreadsheet interface with a standard set of values. In still further embodiments, the user is able to edit the standard set of values so that subsequent selections of the project type will be automatically pre-filled with the updated values.

[00119] Figure 12 depicts an alternate graphical representation of a user interface 1300 for applying volumetric and target metric data of an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00120] In the depicted embodiment, user interface 1300 is an exemplary graphical representation of a user interface within integrated design application 100. Exemplary user interface 1300 may include toolbar 501, panel 503, viewing window 505, exemplary design layout (e.g., layout window 507), File Browser 511, File Information, as previously described. User interface 1300 also includes metric application window 1301.

[00121] User interface 1300 includes metric application window 1301 to allow the application of a volumetric and performance metric to an object selected in layout window 507. Metric application window 1301 includes volumetric selection field 1303, performance metric field 1305, associated metric display 1307, metric definition field 1309 and action buttons (e.g., add button 1311, delete button 1313 and finish button 1315).

[00122] Volumetric selection field 1303 allows the application of volumetric data to the selected object in layout window 507. The user is able to apply a volumetric or performance metric by selecting the name of a pre-defined volumetric in volumetric selection field 1303. In one embodiment, volumetric

selection field 1303 displays all volumetrics defined for the selected object in the composite application design. In other embodiments, volumetric selection field 1303 may also display volumetrics and performance metrics for all related objects (e.g., objects identified as descendants, ancestors or siblings of the selected object), and/or for an entire group of objects of which the selected object belongs.

[00123] Performance metric selection field 1305 allows the application of performance metric data to the selected object in layout window 507. The user is able to apply a performance metric or performance metric by selecting the name of a pre-defined performance metric in performance metric selection field 1305. In one embodiment, performance metric selection field 1305 displays all performance metrics defined for the selected object in the composite application design. In other embodiments, performance metric selection field 1305 may also display performance metrics and performance metrics for all related objects (e.g., objects identified as descendants, ancestors or siblings of the selected object), and/or for an entire group of objects of which the selected object belongs.

[00124] Associated metric display 1307 displays a list of all metrics currently associated with the object selected in layout window 507. In one embodiment, if a volumetric is currently selected by a user in user interface 1301, only volumetrics associated with the object selected in layout window 507 are displayed in associate metric display 1307. Likewise, if a performance metric is

currently selected by a user in user interface 1301, only performance metrics associated with the object selected in layout window 507 are displayed in associate metric display 1307.

[00125] Metric definition field 1309 allows a user to add or edit data corresponding to a volumetric or performance metric selected in volumetric selection field 1301 and performance metric selection field 1303. In one embodiment, only one volumetric or performance metric may be selected in either volumetric selection field 1301 and performance metric selection field 1303.

[00126] Action buttons (e.g., add button 1311, delete button 1313 and finish button 1315) allow a user to perform an action with the entered data. Add button 1309 allows a user to add (apply) a selected volumetric or performance metric to the selected object. Delete button 1311 allows a user to delete (disassociate) a selected volumetric or performance metric to the selected object. Finish button 1315 submits the data and actions performed in user interface 1301 and exits user interface 1301.

[00127] Figure 13 depicts a graphical representation of an exemplary performance file 1400 corresponding to a selected object in an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00128] As depicted in Figure 13, the performance file is implemented as a spreadsheet comparing the performance data entered for the object with the volumetric load data entered for the object. The performance data entered for an object may be entered via, for example, user interface 1200.

[00129] Figure 14 depicts a graphical representation of an exemplary comparison interface 1500 corresponding to volumetric load data and performance metric data of a selected object in an underlying infrastructure during a process of designing a composite application in an integrated design application system, in accordance with one embodiment.

[00130] As depicted in Figure 14, the data comprising a performance file of the selected object is parsed and analyzed to compare volumetric load data of the selected object to the corresponding performance metric data of the selected object. In one embodiment, the comparison is displayed graphically (e.g., the line graph of Figure 14). In alternate embodiments, other visualizations may be available.

[00131] Figure 15 depicts a graphical representation of an exemplary Object Information Window 1600 corresponding to a selected object in a layout of an underlying infrastructure during a process of designing a composite

application in an integrated design application system, in accordance with one embodiment.

[00132] As shown in Figure 15, a typical configuration of Object Information Window 1600 includes Object Information Display 515 and Metric and Volumetric Display 517. Both Object Information Display 515 and Metric and Volumetric Display 517 have been herein described. In one embodiment, Metric and Volumetric Display 517 displays a comparison of the volumetric load data of the selected object to the corresponding performance metric data of the selected object. The data represented in Figure 15 may, for example, be parsed from the performance file of the selected object according to Figure 13, and/or the volumetric and metric data input from the user (or obtained from storage) according to Figures 11 and 12.

[00133] The preceding detailed description includes references to various terminology including, for example, collaborate design application, composite application, system objectives, correlating material, disparate contributors, and information technology objects. For the purposes of clarity, the following are provided as examples only. A collaborative design application may, in various embodiments, comprise and/or be implemented as a compiler, a source code generator, one or more graphical user interfaces, and a software application repository, for instance. A composite application may, in various embodiments, comprise and/or be implemented as one or more of a plurality of executables,

software programs, software packages, data structures, file systems and configuration data. System objectives may include one or more from the group of: performance, efficiency, throughput, bandwidth and storage requirements. Correlating material may include merging program functionality, shared data and/or resolving software or data conflicts. Disparate contributors may be separate (i.e., geographically remote) software developers or designers or separate computer systems that generate computer executable instructions. Information technology tools may include any of the computer software programs, packages, data structures, file systems, binaries, libraries and configuration data, and computing hardware components, devices and the like comprising the architecture of a distributed computing system.

[00134] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

CLAIMS

What is claimed is:

1. A process of designing a composite application system, the method comprising:

defining a plurality of case scenarios performed by a composite application system in a composite application system design;

integrating the plurality of case scenarios with a plurality of execution tools;

specifying an underlying infrastructure for the system design to embody the plurality of execution tools to select a plurality of components whose combination meets a plurality of application requirements, wherein specifying the underlying infrastructure comprises:

i) estimating a performance of the underlying infrastructure according to the composite application system design;

ii) evaluating a feasibility of the composite application system design; and

iii) identifying potential flaws of a feasible composite application system design, and

implementing the composite application system;

wherein the process is performed in an integrated software application executed in a computing system.

2. The process of claim 1, wherein the process is performed over a plurality of design layers.

3. The process of claim 1, wherein the process further comprises:

providing running estimates of metric data, wherein the metric data is updated to correspond to modifications to the composite application system design.

4. The process of claim 3, wherein providing running estimates of metric data comprises combining metric data of the composite application system design from a plurality of design layers.

5. The process of claim 1, wherein the case scenario performed by the composite application system comprises a use case scenario.

6. The process of claim 5, wherein the use case scenario comprises a technical process performed by a component comprising the composite application system design.

7. The process of claim 1, wherein the execution tool comprises a software platform.

8. The process of claim 1, wherein the execution tool comprises a hardware device.

9. The process of claim 1, wherein defining a case scenario for the composite

application system design comprises comparing a target performance for the case scenario to an estimated performance for the case scenario.

10. The process of claim 1, wherein integrating the plurality of case scenarios with a plurality of execution tools comprises:

defining an information technology tool for executing a case scenario; and

mapping the use case scenario to the information technology tool that executes the use case scenario.

11. The process of claim 1, wherein evaluating a feasibility of the composite application system design comprises comparing the composite application system design to user-specified thresholds and computations.

12. The process of claim 11, wherein comparing the composite application system design comprises displaying estimated metrics of the composite application system design with a set of target metrics in an associated programmable platform.

13. The process of claim 1, further comprising correcting a nonconforming composite application system design to conform to the application requirement.

14. The process of claim 13, wherein correcting the nonconforming composite application system design comprises:

visualizing relationships between the plurality of application requirements and a

performance of the underlying infrastructure; and

adjusting the composite application system design to conform with the plurality of application requirements based on visualized relationships.

15. The process of claim 14, wherein adjusting the composite application system design comprises:

estimating composite business metrics; and

modifying a provisioning of the underlying infrastructure based on estimated composite business metrics.

16. The process of claim 1, wherein specifying an underlying infrastructure for the composite application system design further comprises

identifying a plurality of relationships between the plurality of case scenarios;

visualizing each of the plurality of relationships; and

mapping each of the plurality of relationships to a corresponding component of the plurality of components in the composite application system design.

17. The process of claim 16, wherein visualizing each of the plurality of relationships further comprises displaying a component having a relationship with the relationship mapped and the components included in the relationship mapped.

18. The process of claim 16, wherein mapping the relation to the plurality of components in the composite application system comprises:

propagating metric data estimates corresponding to a first component to a second component; and

propagating metric data estimates corresponding to the first component to a third component,

wherein, the second component comprises an identified descendant of the first component, and

wherein the third component comprises an identified ancestor of the first component.

19. The process of claim 1, further comprising:

providing collaboration among distributed users of the integrated software application, wherein, providing collaboration among distributed users of the integrated software application comprises allowing direct simultaneous collaboration performed in the integrated software application.

20. The process of claim 19, further comprising

communicating with other contributors via messaging performed in the integrated software application, wherein

messaging comprises attaching messages to objects in the composite application system design.

21. The process of claim 19, wherein client devices of users of the integrated software application are connected to client devices of other users of the integrated

software application via a network.

22. The process of claim 21, wherein the network is a peer-to-peer network.

23. The process of claim 22, wherein a client device of a user of the integrated software application operates as a peer in the peer-to-peer network.

24. The process of claim 21, wherein the network is a client-server network.

25. The process of claim 24, wherein the client device of a user of the integrated software application operates as a client in the client-server network.

26. The process of claim 24, wherein the integrated software application further comprises a central repository that operates as a server in the client-server network.

27. The process of claim 19, wherein providing collaboration among distributed users of the integrated software application further comprises tracking modifications to the composite application system by contributors.

28. The process of claim 19, wherein providing collaboration among distributed users of the integrated software application further comprises maintaining version control of distributed instances of the design of the composite application system.

29. The process of claim 28, wherein maintaining version control comprises:

automatically updating documentation of the composite application system;

ensuring exclusive access to files; and

providing merging functionality.

30. The process of claim 19, wherein providing collaboration among distributed users of the integrated software application comprises archiving a version of a composite application system design.

31. The process of claim 1, further comprising providing the capability to leverage pre-existing material, wherein, providing the capability to leverage pre-existing material comprises:

referencing a repository for storing pre-designed composite application systems for use as templates.

32. The process of claim 1, further comprising:

designing the composite application system based on the underlying infrastructure, the composite application system comprising a plurality of components distributed over a plurality of domains; and

displaying the composite application system over the plurality of domains; and

providing estimated metric data for the composite application system, wherein providing estimated metric data comprises combining metric data for each of the

plurality of domains comprising the composite application system.

33. The process of claim 1, further comprising:

relating composite metrics for the composite application system, the composite metrics comprising an aggregate of a plurality of metrics corresponding to the plurality of components comprising the composite application system; and

incorporating applicable project type details, the project type details comprising a plurality of pre-defined data corresponding to a project type selected by a user for the composite application system,

wherein relating composite metrics are performed on-demand.

34. The process of claim 1, further comprising providing estimated calculations of metric data for the composite application system; wherein providing estimated calculations of the metric data comprises:

providing running estimates of metric data throughout the design process;

updating the estimates of metric data to correspond to modifications of the composite application system design.

35. The process of claim 1, further comprising debugging the composite application system design which does not meet the plurality of application requirements, wherein debugging a composite application system design comprises:

graphically displaying attributes of the composite application system design which do not conform to the plurality of application requirements; and

adjusting a provisioning of components of the composite application system design to align with the plurality of application requirements.

36. The process of claim 35, wherein graphically displaying estimated attributes of the composite application system design further comprises displaying a comparison between estimated attributes of the composite application system design with the plurality of application requirements.

37. The process of claim 36, wherein displaying a comparison between the estimated attributes of the composite application system design with the plurality of application requirements further comprises:

adjusting a displayed comparison between the estimated attributes of the composite application system design with the plurality of application requirements to account for an adjusted provisioning of components of the composite application system design,

wherein, adjusting the displayed comparison is performed in real time.

38. A computer-readable medium having computer executable instructions stored thereon for execution by one or more computers, that when executed performs a process of simulating the performance of a composite application system subject to a plurality of application requirements, the process comprising:

receiving an input defining a plurality of case scenarios performed by a composite application system in a system design;

integrating the plurality of case scenarios with a plurality of components of the composite application system, the plurality of components comprising a collection of software applications and hardware components;

receiving an input specifying an underlying infrastructure for the system design to embody the plurality of components, wherein specifying the underlying infrastructure includes specifically selecting a plurality of software applications and hardware components corresponding to the collection of components, wherein the plurality of software applications and hardware components meets the plurality of application requirements, and wherein specifying the underlying infrastructure further comprises:

estimating a performance of the underlying infrastructure according to the system design by aggregating the user inputted estimated performance of each of the components in the composite application system design, wherein the estimated performance is evaluated by comparing said estimated performance with system performance requirements, which are user-specified by thresholds and computations and if the estimated performance is larger than or equal to the system performance requirements, then the estimated cost of the composite application system design is further compared with system budget requirements; and

identifying potential flaws of feasible designs by determining the specific components that cause the system design to fail to meet system requirements by comparing the estimated performance and estimated cost of each component in the composite application system design with an expected performance and expected cost of a component that meets system

requirements,

further wherein the process is performed by an integrated software application, wherein the composite application system design is implemented as one or more of a plurality of executables, software programs, software packages, data structures, file systems and configuration data.

39. The computer-readable medium of claim 38, wherein the process further comprises:

providing running estimates of metric data, wherein the metric data is updated to correspond to modifications to the system design.

40. The computer-readable medium of claim 39, wherein providing running estimates of metric data comprises combining metric data of the system design from a plurality of design layers.

41. The computer-readable medium of claim 38, wherein defining a case scenario for the system design comprises comparing a target performance for the case scenario to an estimated performance for the case scenario.

42. The computer-readable medium of claim 38, wherein integrating the plurality of case scenarios with a plurality of execution tools comprises:

defining an information technology tool for executing a case scenario; and

mapping the use case scenario to the information technology tool that executes

the use case scenario.

43. The computer-readable medium of claim 42, wherein specifying an underlying infrastructure for the system design further comprises

identifying a plurality of relationships between the plurality of components;

visualizing each of the plurality of relationships; and

mapping each of the plurality of relationships to a corresponding component of the plurality of components in the system design.

44. The computer-readable medium of claim 43, wherein mapping the relation to the plurality of components in the composite application system comprises:

propagating metric data estimates corresponding to a first component to a second component; and

propagating metric data estimates corresponding to the first component to a third component,

wherein, the second component comprises an identified descendant of the first component, and

wherein the third component comprises an identified ancestor of the first component.

45. The computer-readable medium of claim 38, further comprising correcting a nonconforming system design to conform to the application requirement, wherein correcting the nonconforming composite application system design

comprises:

visualizing relationships between the plurality of application requirements and a performance of the underlying infrastructure; and

adjusting the system design to conform with the plurality of application requirements based on visualized relationships.

46. The computer-readable medium of claim 38, wherein, the integrated software application is operable to be configured to provide collaboration among distributed users of the integrated software application, wherein, providing collaboration among distributed users of the integrated software application comprises enabling direct simultaneous collaboration performed in the integrated software application.

47. A system for implementing a collaborative design application for simulating the performance of a composite application system according to a system objective, the collaborative design application comprising:

an interface for receiving an input defining a plurality of case scenarios performed by a composite application system in a system design and for receiving an input specifying an underlying infrastructure for the system design to embody the plurality of components, wherein specifying the underlying infrastructure includes specifically selecting a plurality of software applications and hardware components corresponding to the collection of components, wherein the plurality of software applications and hardware components meets the plurality of application requirements;

a repository, communicatively coupled to the interface, for storing completed composite application system designs;

a database, communicatively coupled to the interface, for collecting data of individual information technology objects according to a plurality of metrics;

a first viewer, coupled to the interface, for displaying volumetric load data and performance metric data of the composite application system design; and

a second viewer, coupled to the interface, for graphically representing relationships between information technology objects in the composite application system design,

wherein the system is implemented as a computer-readable medium storing a computer-executable integrated software application, the integrated software application when executed by at least one processor causing the at least one processor to:

simulate the performance of the composite application system comprising a plurality of components comprising a collection of software applications and hardware components, and

integrate the plurality of case scenarios with a plurality of components of the composite application system, and wherein specifying the underlying infrastructure further comprises:

estimating a performance of the underlying infrastructure according to the system design by aggregating the user inputted estimated performance of each of the components in the composite application system design,

wherein the estimated performance is evaluated by comparing said estimated performance with system performance requirements, which are user-specified by thresholds and computations and if the estimated performance is larger than or equal to the system performance requirements, then the estimated cost of the composite application system design is further compared with system budget requirements;

identify potential flaws of feasible designs by determining the specific components which cause the system design to fail to meet system requirements by comparing the estimated performance and estimated cost of each component in the composite application system design with an expected performance and expected cost of a component that meets system requirements

wherein further, the design of the composite application system comprises the input of a plurality of disparate contributors received from the interface.

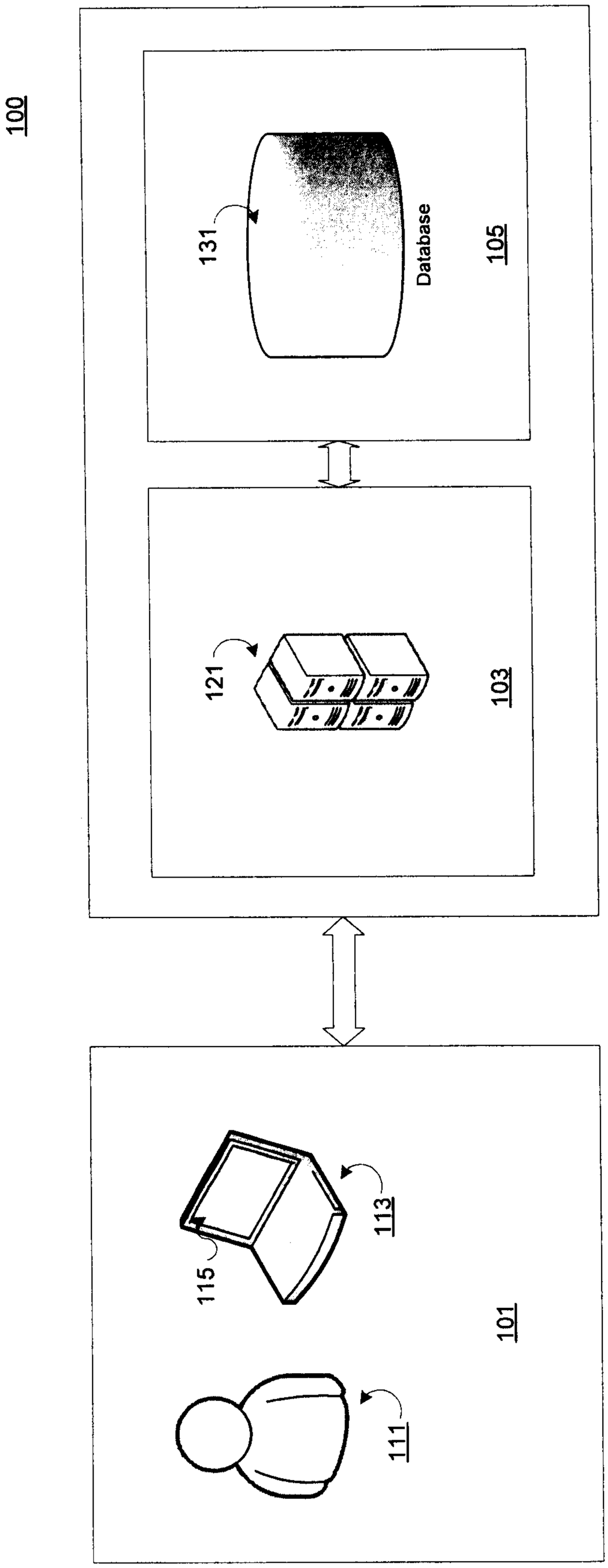
48. The system of claim 47, wherein the collaborative design application obtains metric data corresponding to information technology objects in the composite application system design from the database.

49. The system of claim 47, wherein volumetric load data corresponding to information technology objects in the composite application system design are compared to target metrics based on a system objective.

50. The system of claim 47, wherein the collaborative design application leverages completed composite application system designs stored in the repository as templates in designing new composite application system designs.

51. The system of claim 47, wherein changes made to the composite application system design by the plurality of disparate contributors are tracked by the collaborative design application.

52. The system of claim 47, wherein the second viewer for graphically representing relationships between the information technology objects in the composite application system design allows users to explicitly create and view relationships across a plurality of views of the composite application system design.



1/15

FIG. 1

Patent Agents
Smart & Biggar

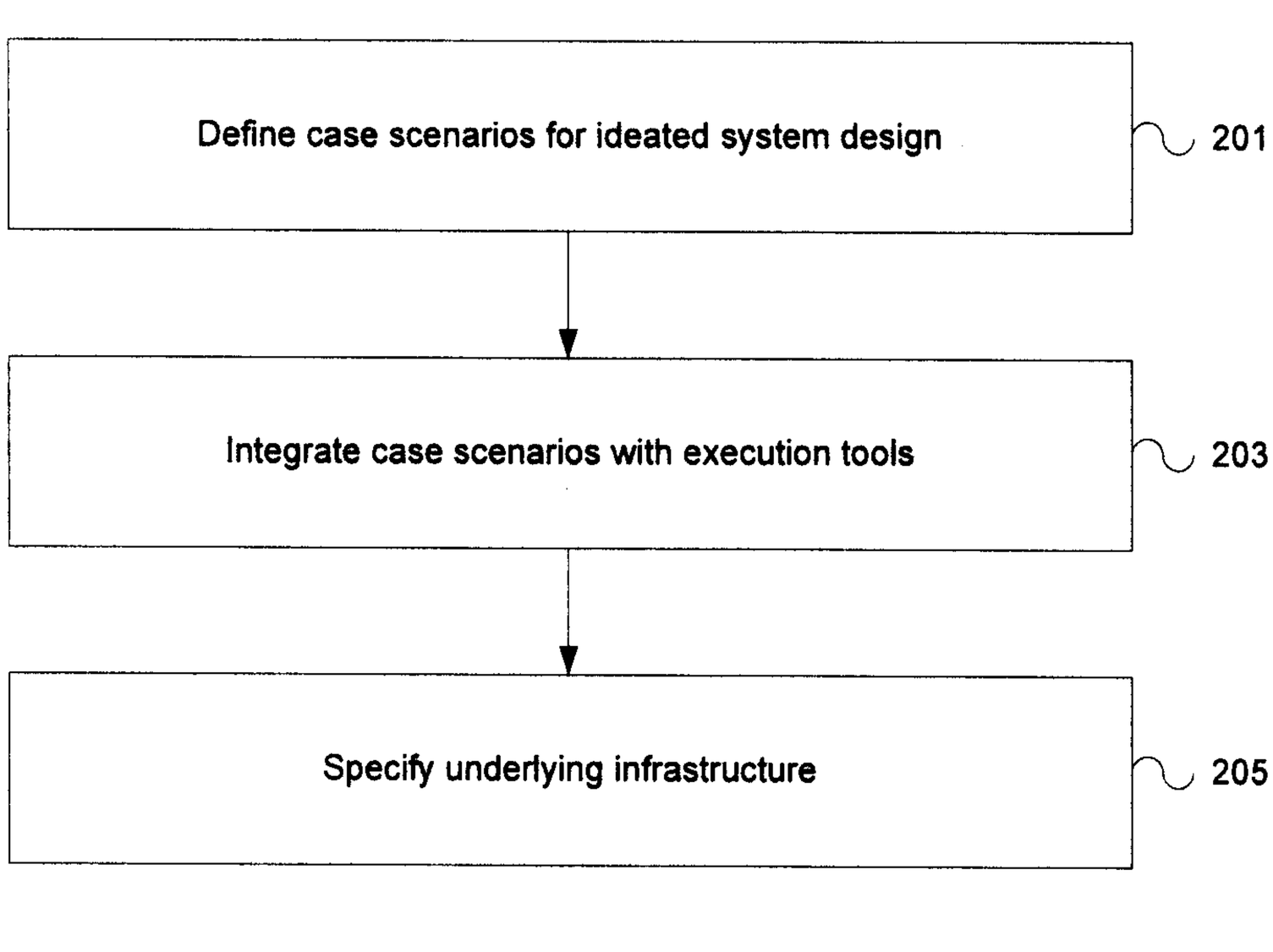

200
Define case scenarios for ideated system design
201
Integrate case scenarios with execution tools
203
Specify underlying infrastructure
205

FIG. 2

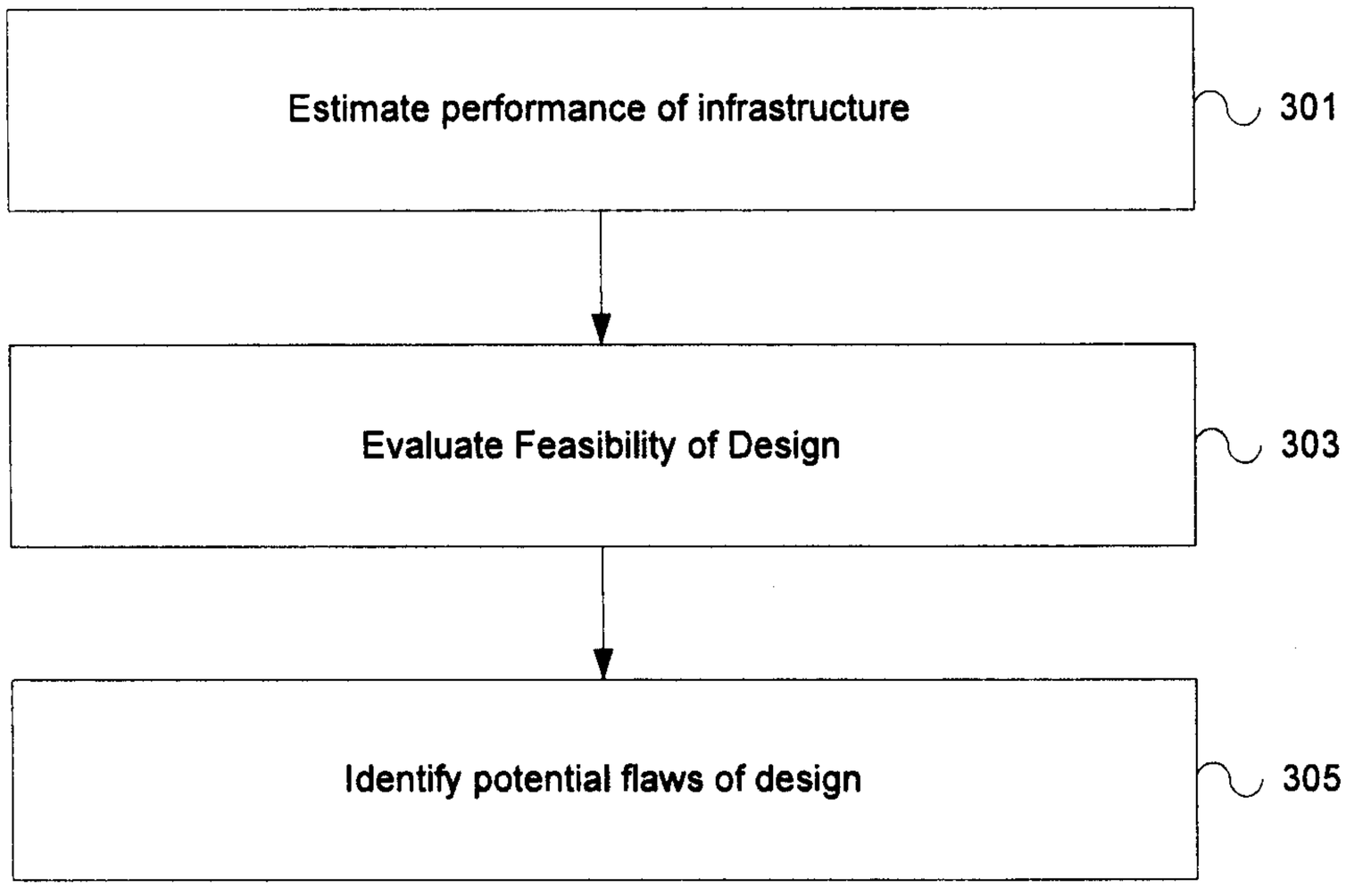
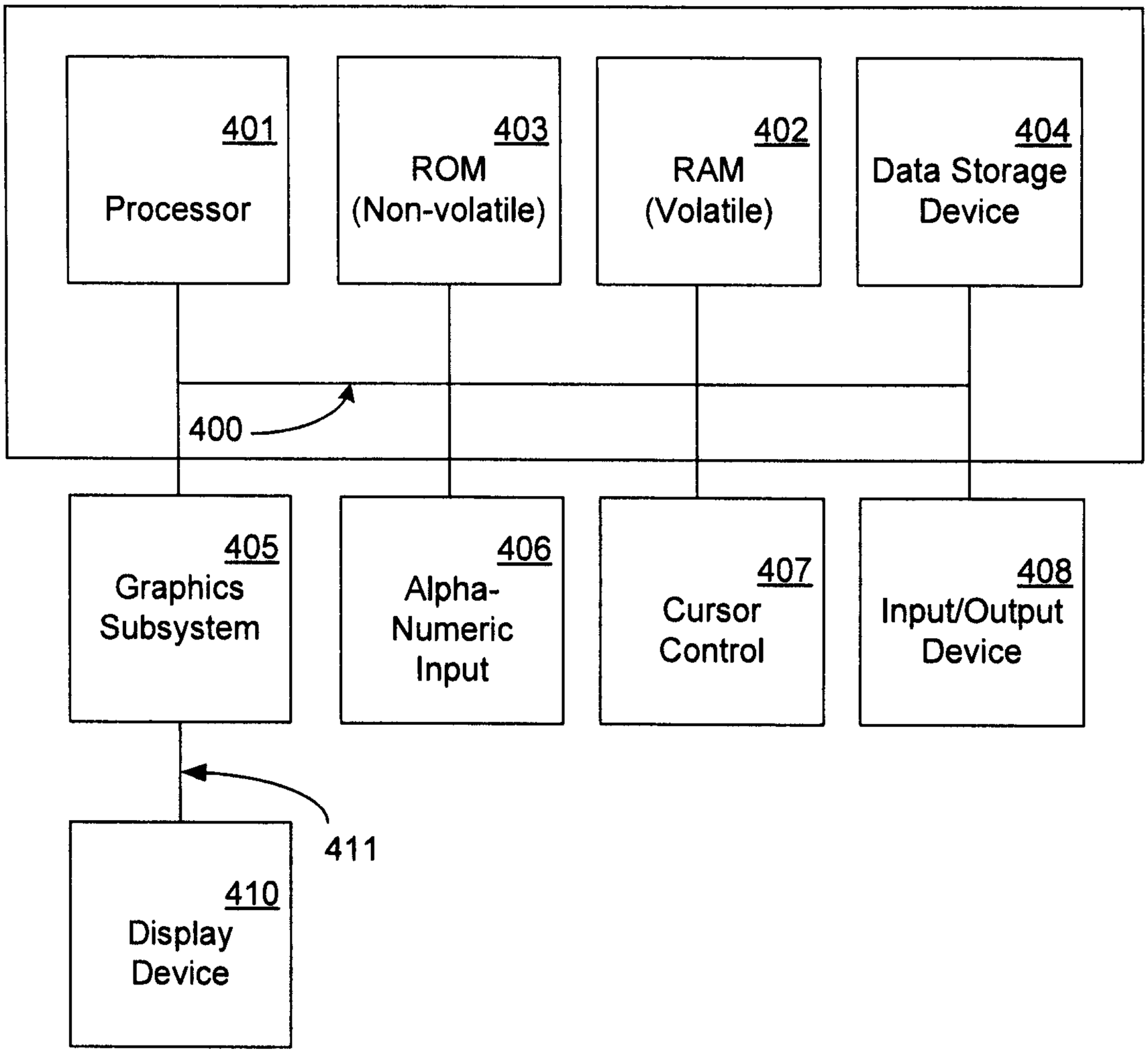

300
Estimate performance of infrastructure
301
Evaluate Feasibility of Design
303
Identify potential flaws of design
305

FIG. 3



System 412

FIG. 4

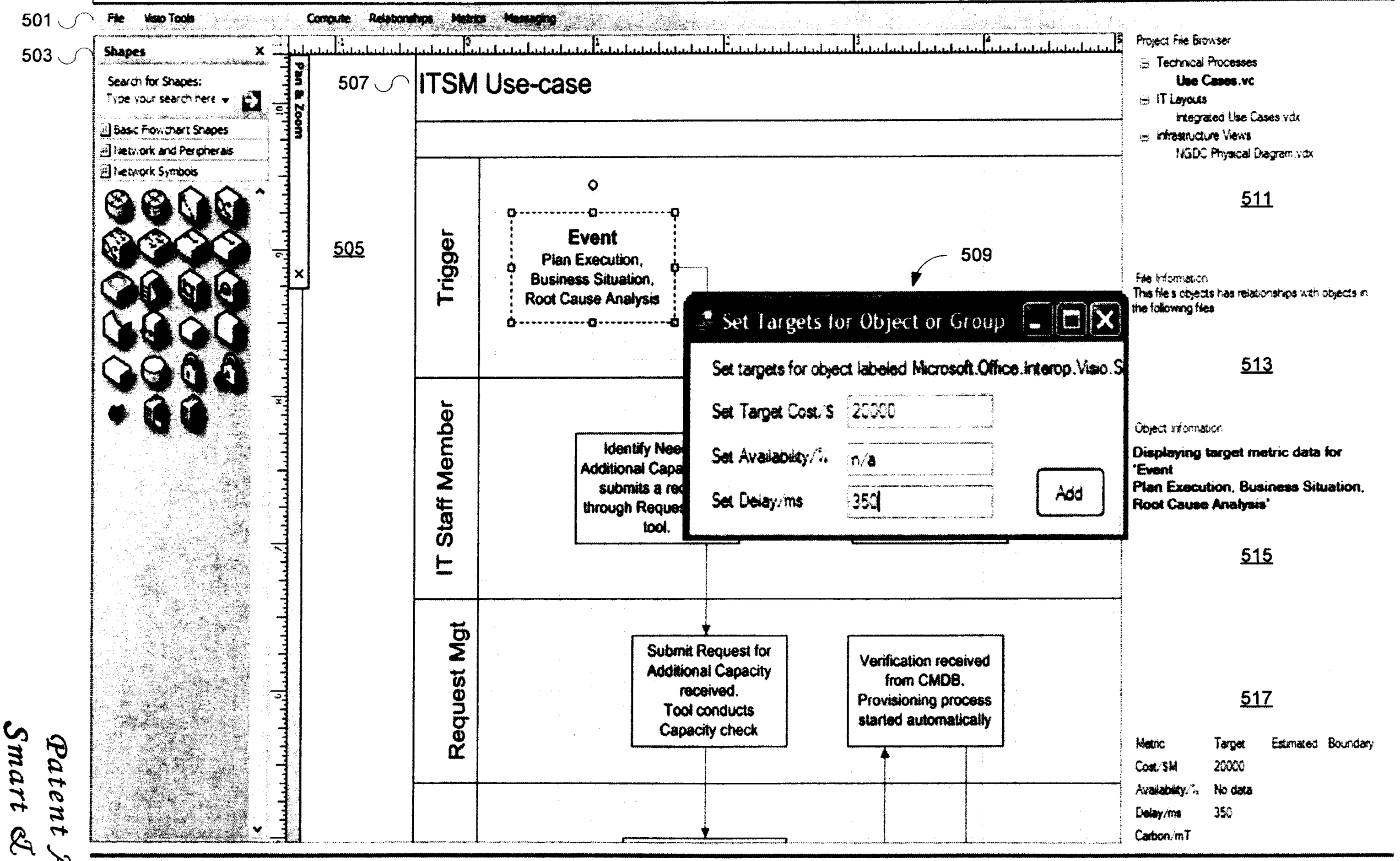


FIG. 5

Patent Agents
Smart & Biggar

CA 02680978 2009-10-02

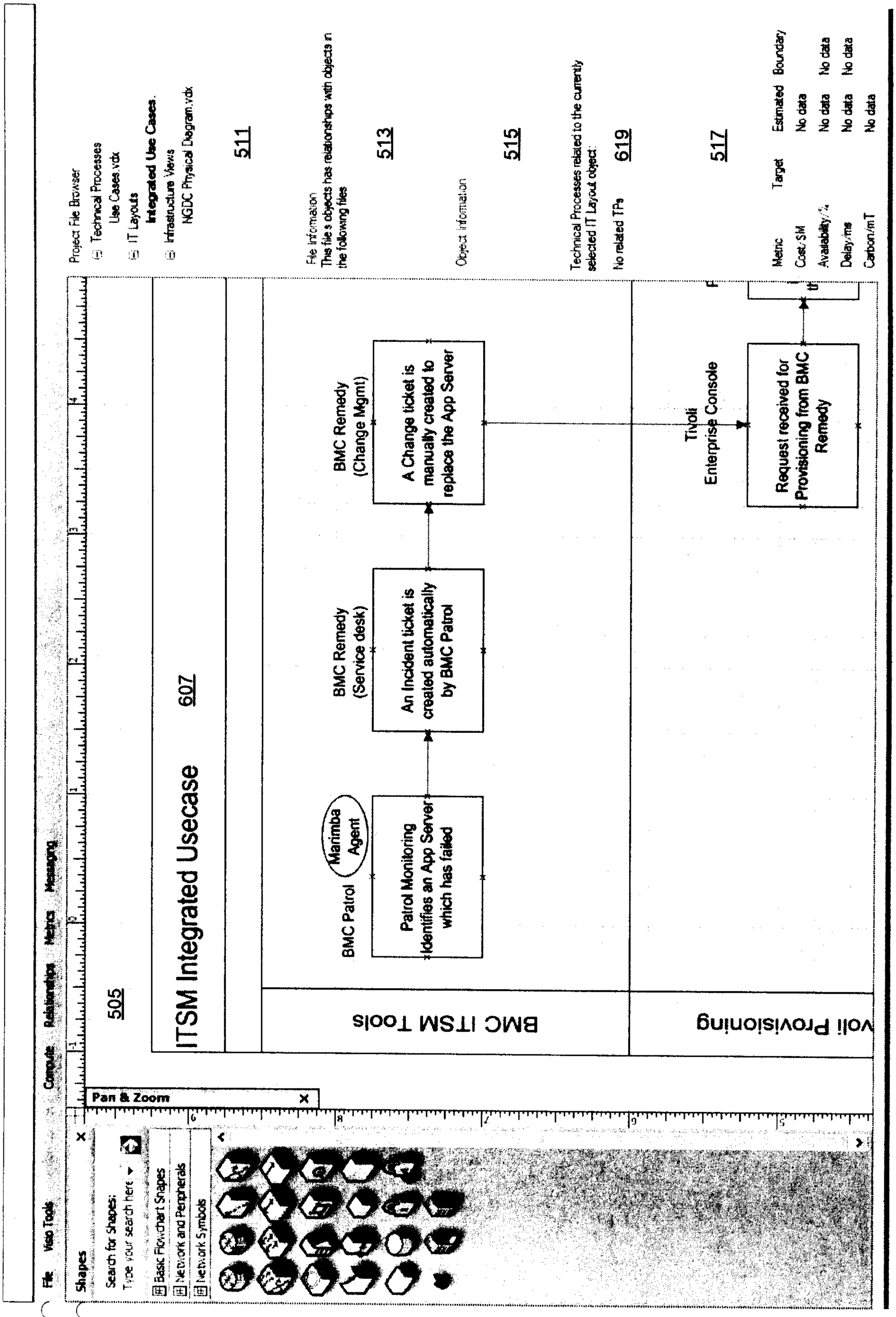


FIG. 6

Patent Agents
Smart & Biggar

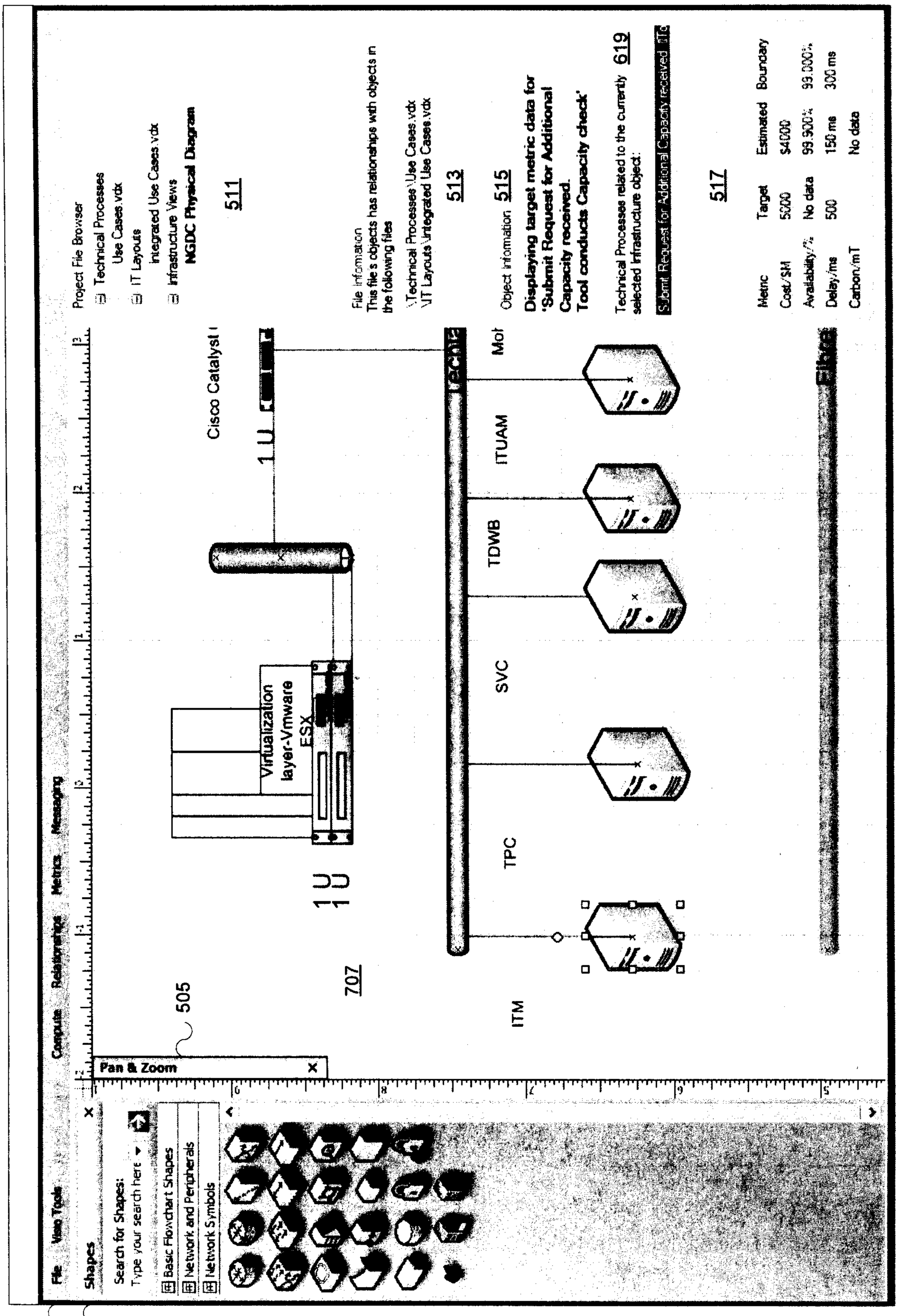


FIG. 7

Agents
L Biggar

800

Relationship Mapper

809

Relevant Elements

Listed here are the tech process elements that may be mapped

ITSM Use-case
:Provisioning
:CMDB
:Request Mgt
IT Staff Member
Trigger
ITSM Use-case
ITSM Use-case
Verification received from C
Additional Capacity notificat
Submit Request for Addition
Update Finance and Billing
Provide Progress Reports
Update CMDB :- Business
IT Staff receives confirmatic
Event:Plan Execution, Busi
:Provisioning request receiv
Notify Provisioning Complet
Monitor Progress through cy
Verify Capacity is Available f
Identify Need for Additional

elect an object from the Technical Process drawing, and then select the corresponding IT Layout Object

803

ITSM Use-case

Trigger

IT Staff Member

Request Mgt

CMDB

Provisioning

Pan & Zoom

807

ITSM Integrated Usecase

BMC ITSM Tools

Tivoli Provisioning

Atrium CMDB

Ordered Relationships

Trigger - Patrol Monitoring Identifies an App Serve
Trigger - An Incident ticket is created automatically
Identify Need for Additional Capacity and submits a
Submit Request for Additional Capacity received

811

Create | Delete | Up | Down

Save and Close

Submit Request for Additional Capacity received.

Request received for Provisioning from BMC Remedy

FIG. 8

Patent Agents
Smart & Biggar

CA 02680978 2009-10-02

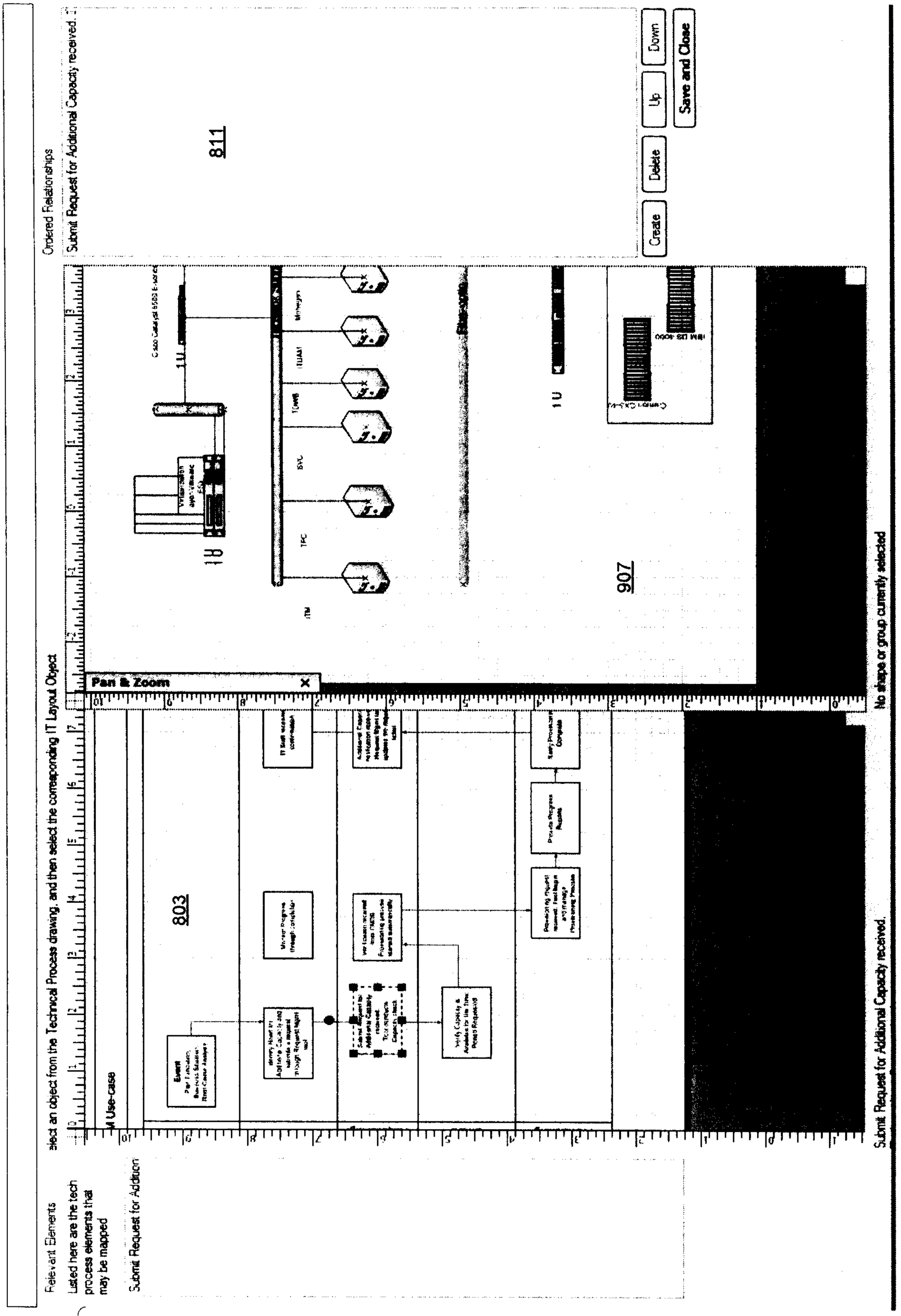


FIG. 9

Patent Agents
Smart & Biggar

FIG. 10A

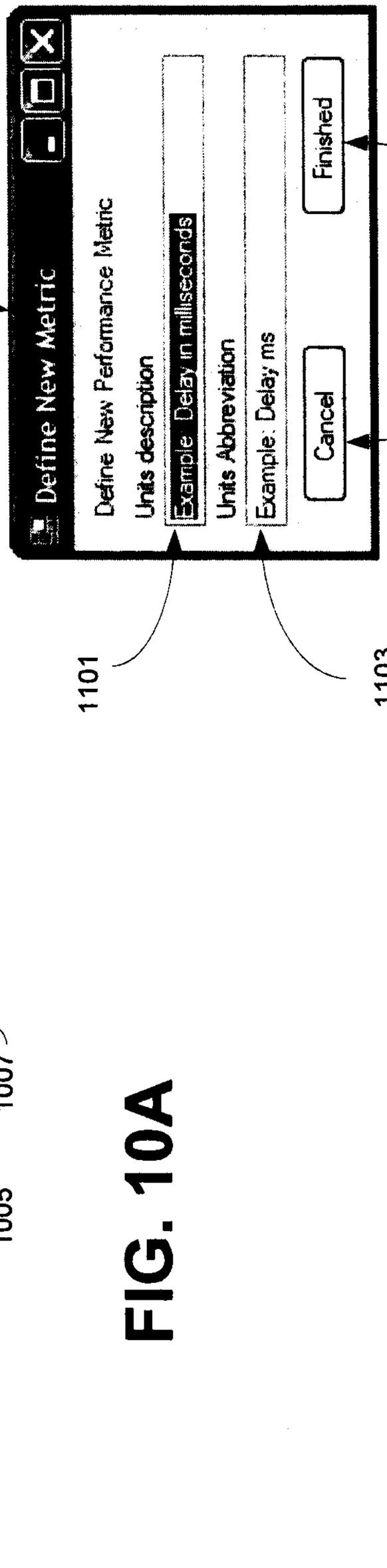


FIG. 10B

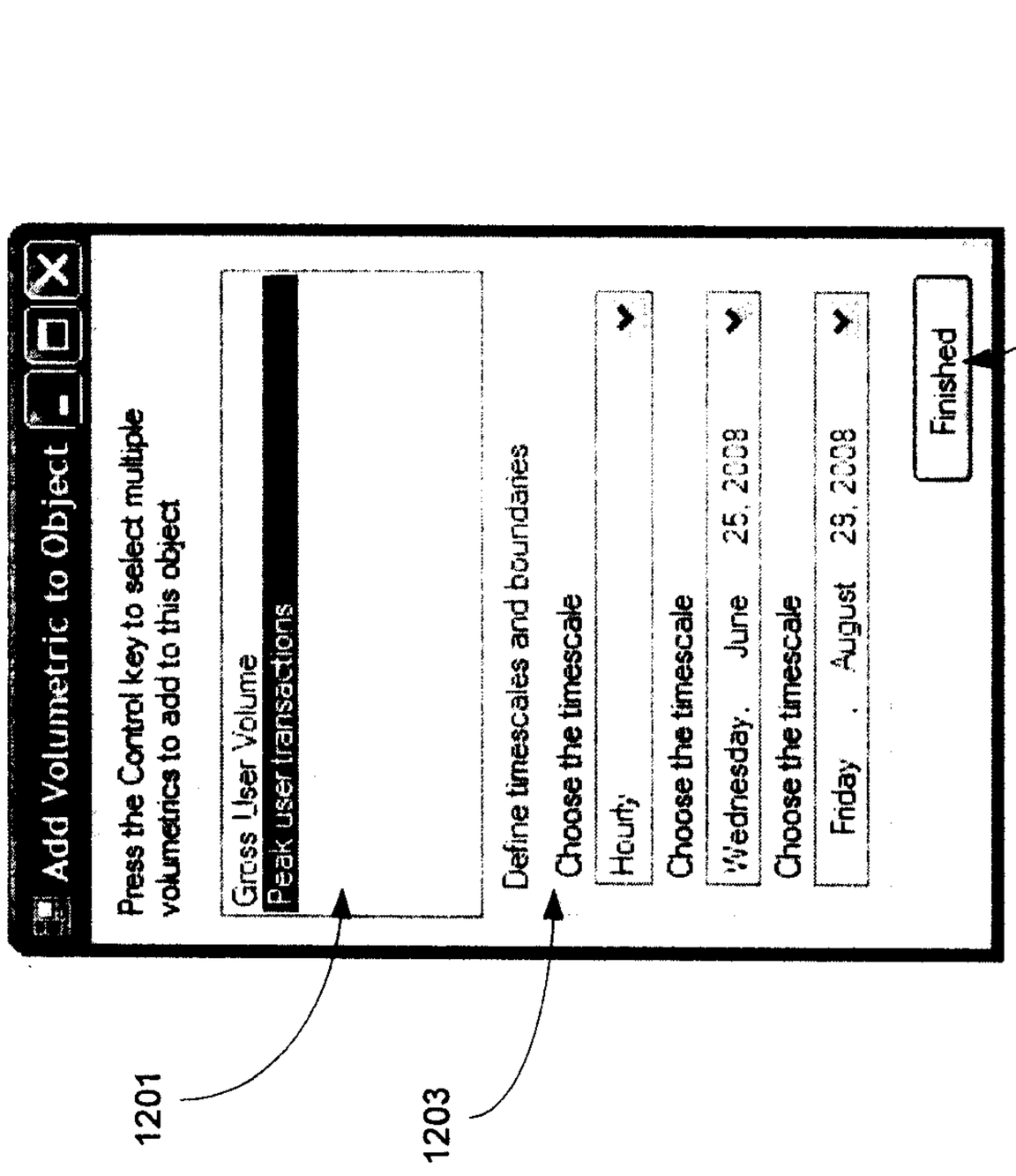
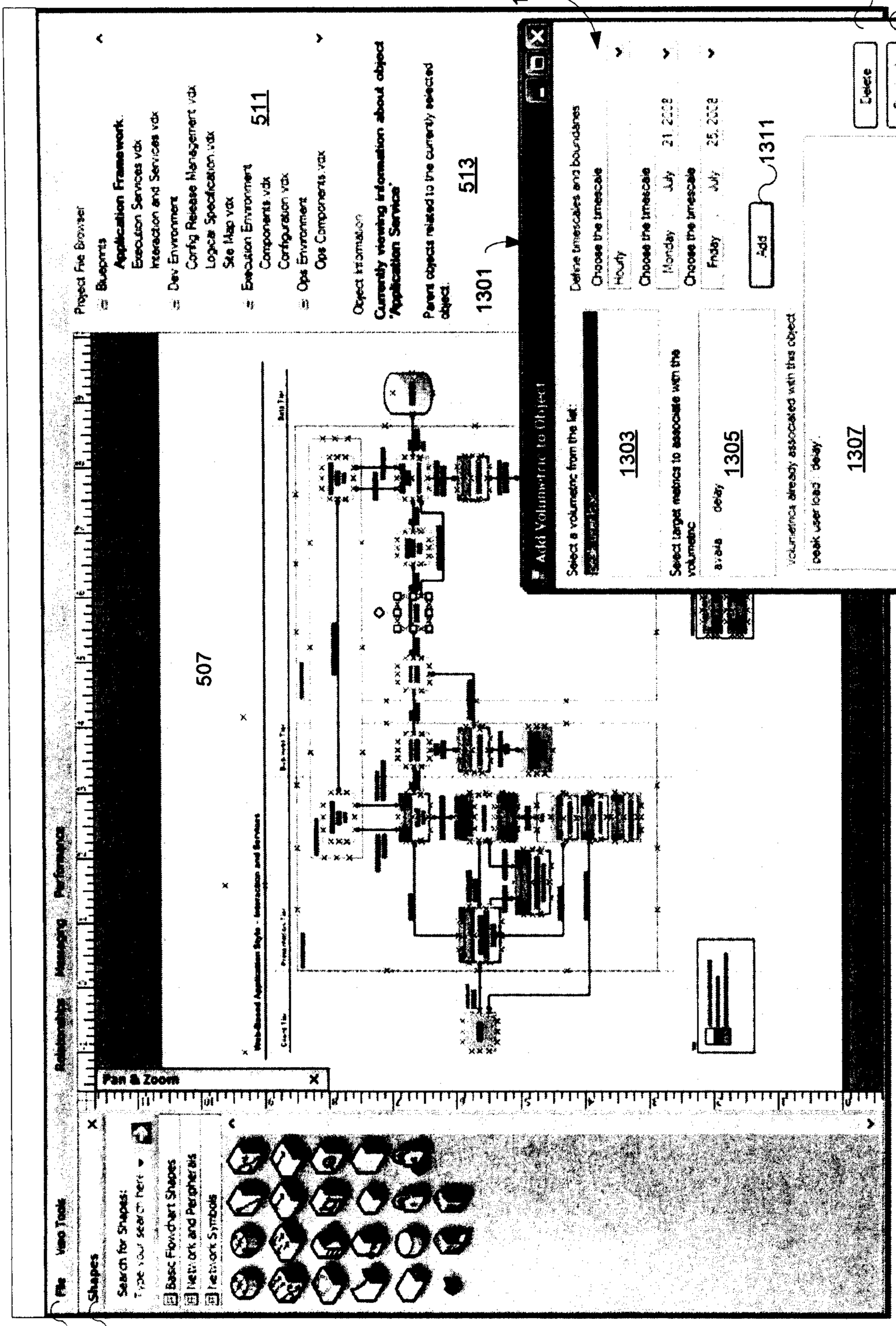

1200
Add Volumetric to Object
Press the Control key to select multiple volumetrics to add to this object
Gross User Volume
Peak user transactions
Define timescales and boundaries
Choose the timescale
Hourly
Choose the timescale
Wednesday, June 25, 2008
Choose the timescale
Friday , August 23, 2008
Finished
1201
1203
1207

FIG. 11



12/15

FIG. 12

Patent Agents
Smart & Biggar

1400

File Edit View Insert Format Tools Data Window Help — Type a question for help

Arial 10 B I U

Reply with Changes... End Review...

H5 7/18/2008 10:38:58 PM

	A	B	C	D	E	F
1	**Description**	**Timescale Unit**	**Measurement Unit**	**Start Date**		
2						
3	**Volumetrics and associated metrics**					
4						
5				7/14/2008 22:38	7/15/2008 22:38	7/16/2008 22:38
6	Number transactions	Daily	#	10000	10000	2000000
7	Availability		%	98	98	99.999
8	delay		ms	500	500	1000
9						
10	**Totalled Volumetric Loads**					
11	**Performance Measure**	**Associated Process**				
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Sheet4 / Caching / HPCClus / **DataServices** / Sheet1 / Sheet2 / Sheet3 /

Ready — Sum=12/16/8435 13:1

FIG. 13

Patent Agents
Smart & Biggar

CA 02680978 2009-10-02

1500

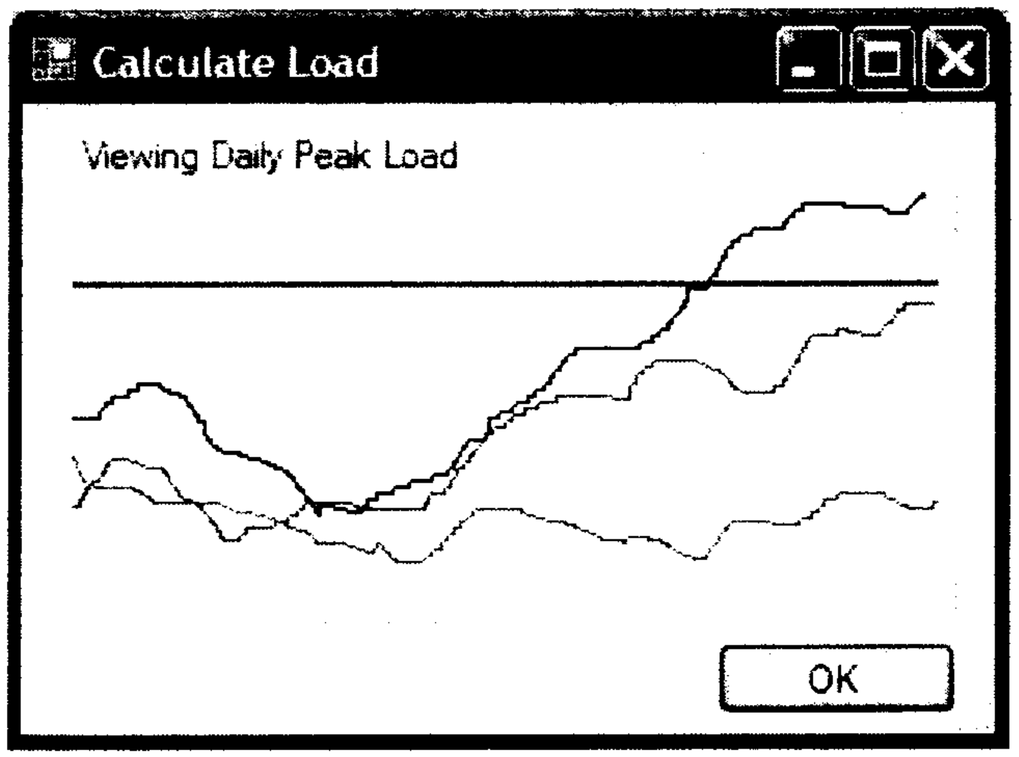


FIG. 14

CA 02680978 2009-10-02

Patent Agents
Smart & Biggar

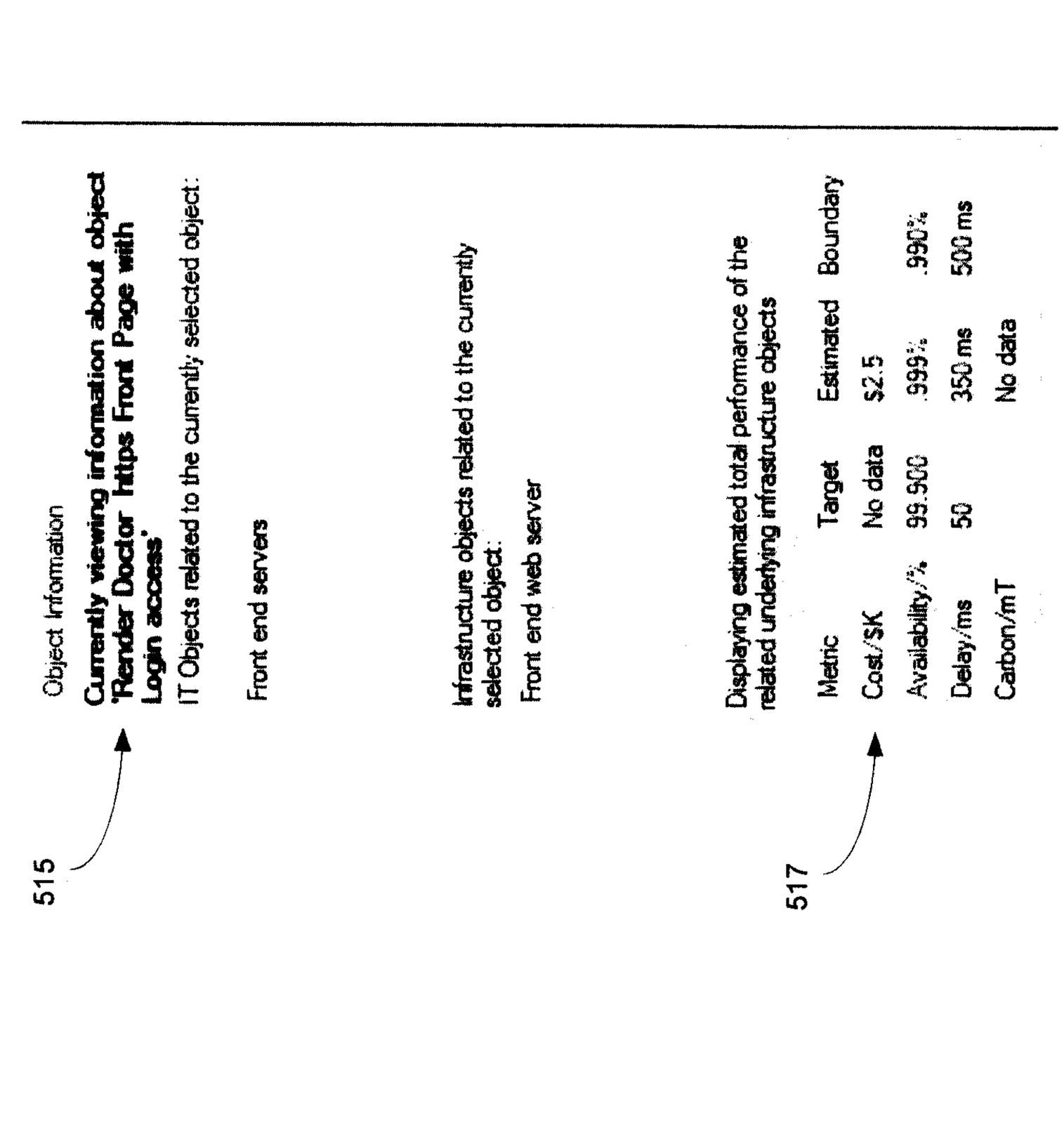
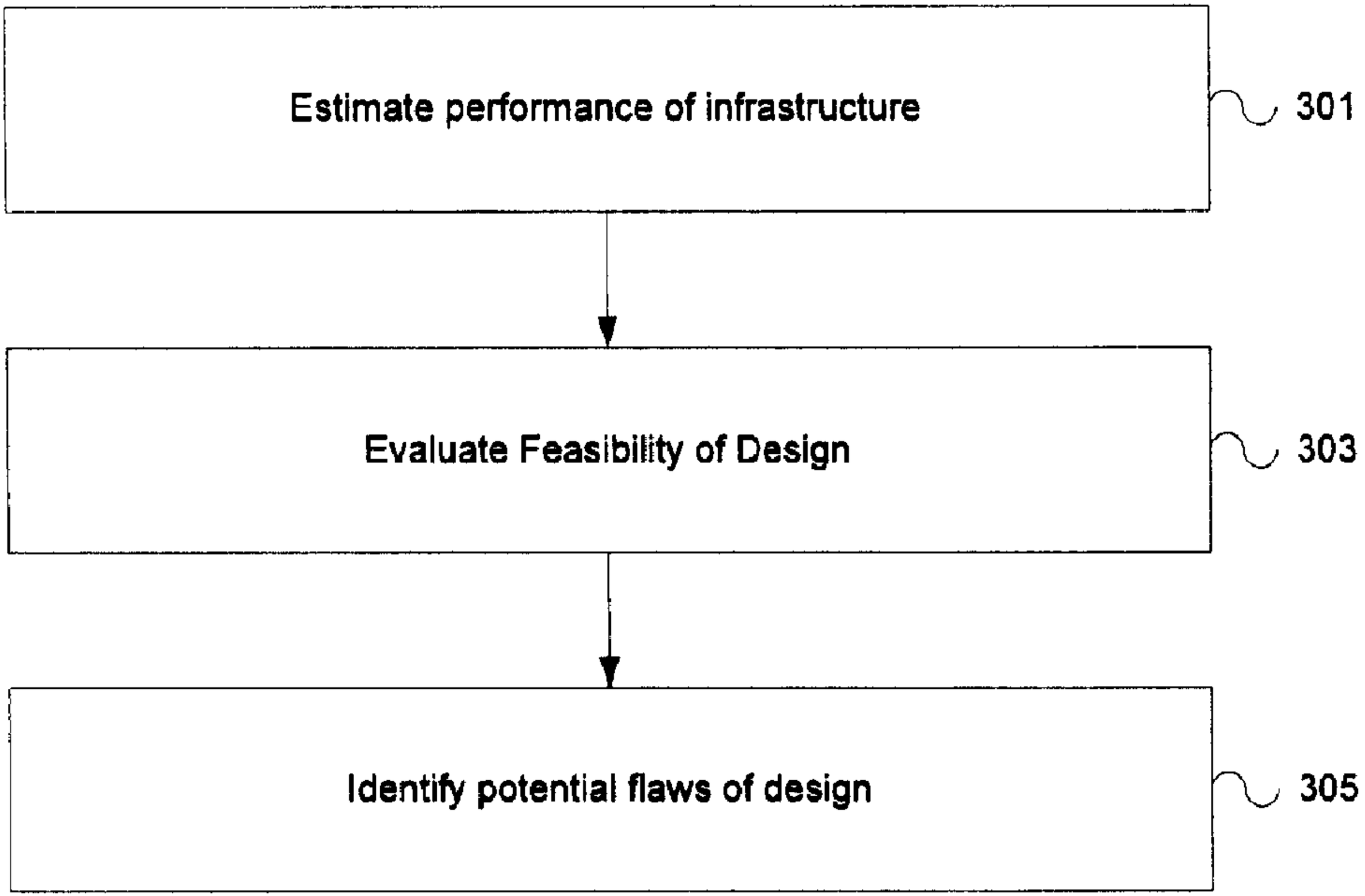


FIG. 15

Patent Agents
Smart & Biggar


Estimate performance of infrastructure
301
Evaluate Feasibility of Design
303
Identify potential flaws of design
305