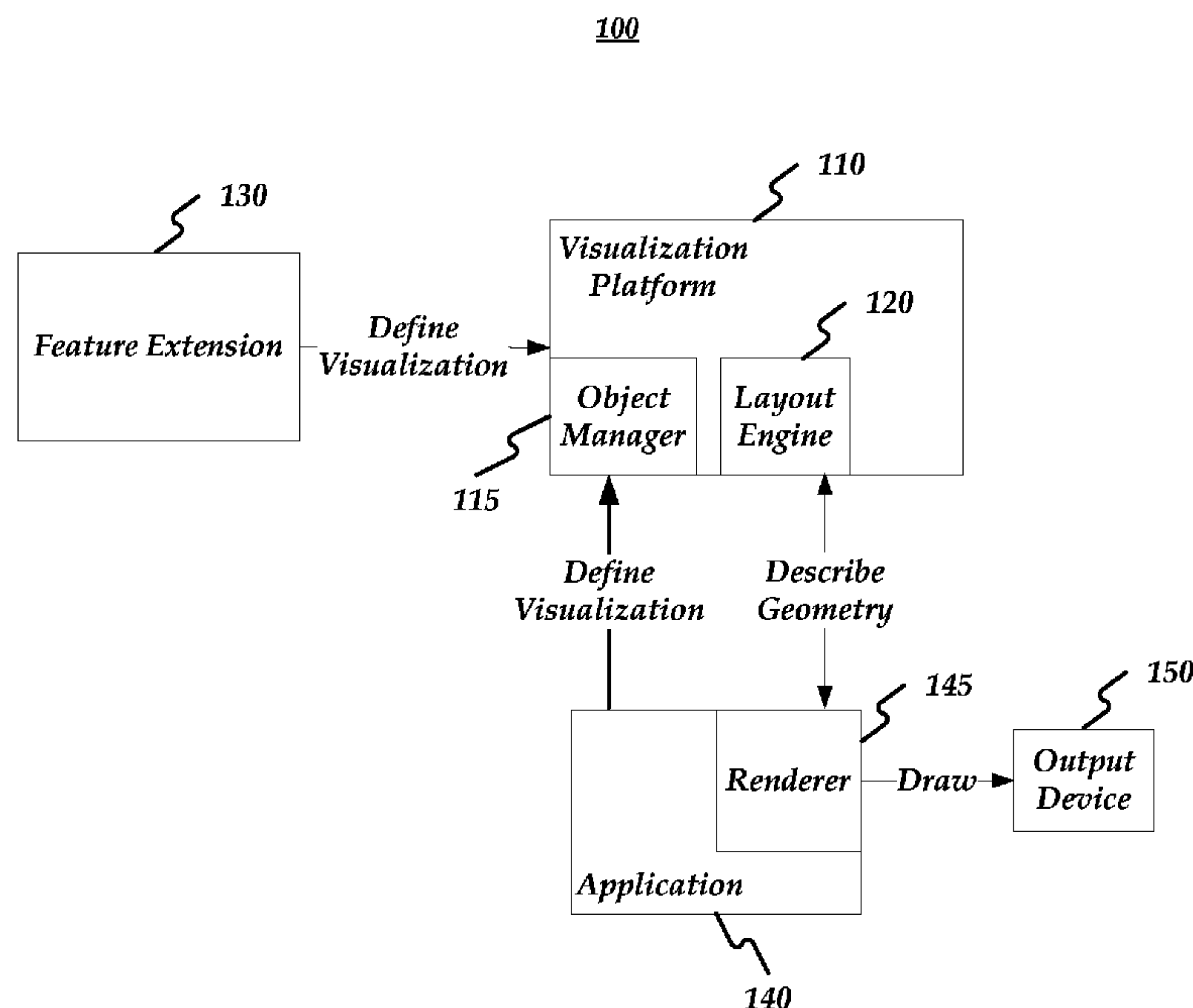




(86) Date de dépôt PCT/PCT Filing Date: 2010/04/23  
(87) Date publication PCT/PCT Publication Date: 2010/11/04  
(85) Entrée phase nationale/National Entry: 2011/09/27  
(86) N° demande PCT/PCT Application No.: US 2010/032308  
(87) N° publication PCT/PCT Publication No.: 2010/126803  
(30) Priorité/Priority: 2009/04/30 (US12/433,327)

(51) Cl.Int./Int.Cl. *G06F 9/44* (2006.01),  
*G06F 17/00* (2006.01), *G06F 3/14* (2006.01)  
(71) Demandeur/Applicant:  
MICROSOFT CORPORATION, US  
(72) Inventeurs/Inventors:  
ALLYN, BARRY CHRISTOPHER, US;  
RUBLE, B. SCOTT, US;  
RAMPSON, BENJAMIN EDWARD, US  
(74) Agent: SMART & BIGGAR

(54) Titre : CADRE D'EXTENSIBILITE DE PLATE-FORME  
(54) Title: PLATFORM EXTENSIBILITY FRAMEWORK



**FIG. 1**

(57) **Abrégé/Abstract:**

A platform extensibility framework may be provided. A visualization platform may register feature extensions and receive requests to create objects. The objects may be created according to drawing instructions provided by the visualization platform, an application requesting the object's creation, and/or at least one of the feature extensions.

## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau

(43) International Publication Date  
4 November 2010 (04.11.2010)



(10) International Publication Number  
**WO 2010/126803 A3**

## (51) International Patent Classification:

*G06F 9/44* (2006.01) *G06F 17/00* (2006.01)  
*G06F 3/14* (2006.01)

## (21) International Application Number:

PCT/US2010/032308

## (22) International Filing Date:

23 April 2010 (23.04.2010)

## (25) Filing Language:

English

## (26) Publication Language:

English

## (30) Priority Data:

12/433,327 30 April 2009 (30.04.2009) US

(71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **ALLYN, Barry Christopher**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **RUBLE, B. Scott**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **RAMPSON, Benjamin Edward**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

## Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

## Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

## (88) Date of publication of the international search report:

3 February 2011

(54) Title: PLATFORM EXTENSIBILITY FRAMEWORK

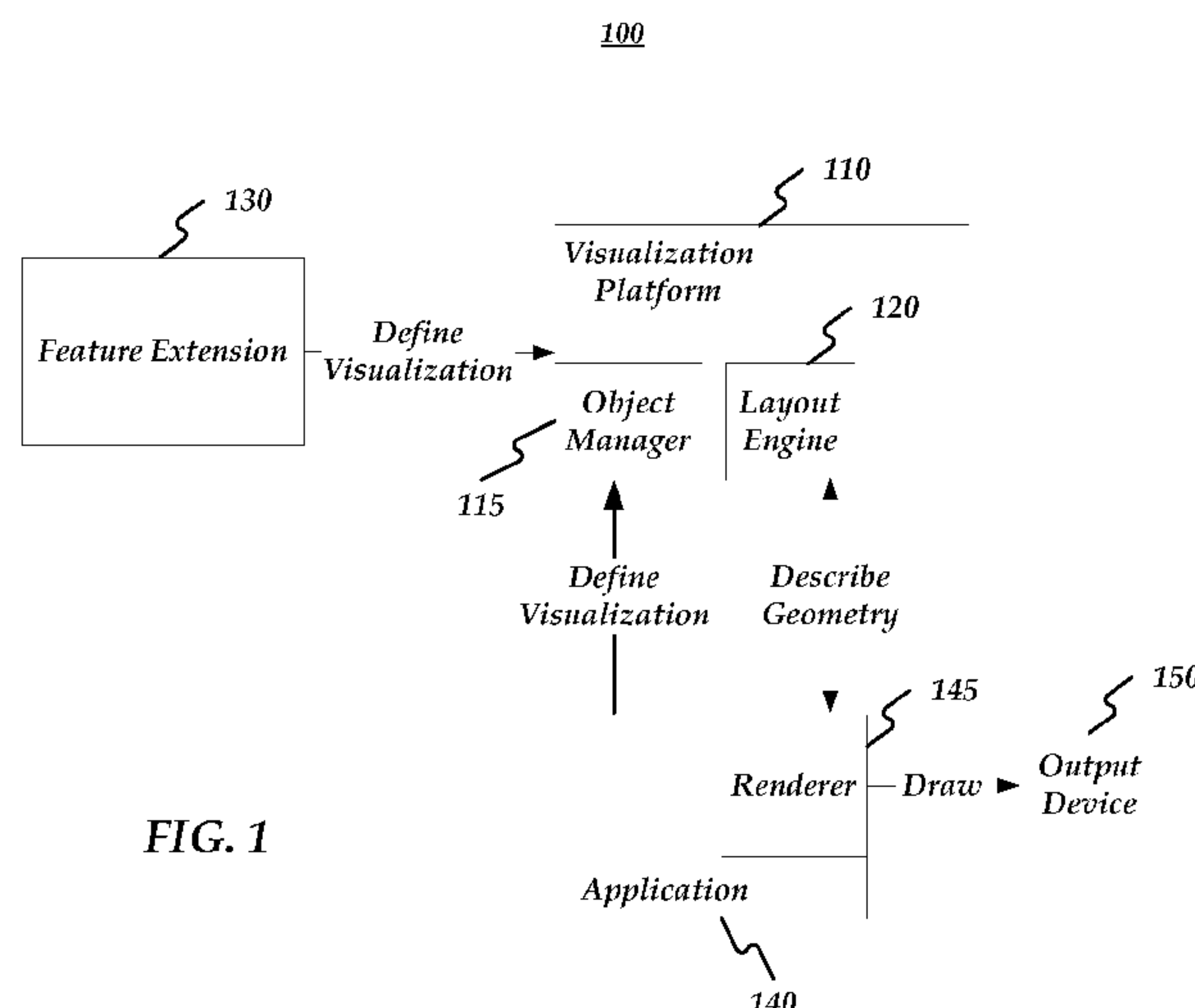


FIG. 1

(57) Abstract: A platform extensibility framework may be provided. A visualization platform may register feature extensions and receive requests to create objects. The objects may be created according to drawing instructions provided by the visualization platform, an application requesting the object's creation, and/or at least one of the feature extensions.

WO 2010/126803 A3



## PLATFORM EXTENSIBILITY FRAMEWORK

### BACKGROUND

[001] Platform extensibility provides a framework for allowing feature plug-ins to communicate with software applications. In some situations, product release schedules  
5 may dictate when new features may be incorporated into an application. For example, a new application version may be released on an annual basis, but users may request new features more frequently. Thus, the conventional strategy is to incorporate selected features in each scheduled release. This often causes problems because the conventional strategy is not always responsive to user demand. For example, a requested feature may  
10 take years to be incorporated into an application release and may result in customer dissatisfaction.

### SUMMARY

[002] A platform extensibility framework may be provided. This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in  
15 the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter. Nor is this Summary intended to be used to limit the claimed subject matter's scope.

[003] A platform extensibility framework may be provided. A visualization platform may register feature extensions and receive requests to create objects. The objects may be  
20 created according to drawing instructions provided by the visualization platform, an application requesting the object's creation, and/or at least one of the feature extensions.

[004] Both the foregoing general description and the following detailed description provide examples and are explanatory only. Accordingly, the foregoing general description and the following detailed description should not be considered to be  
25 restrictive. Further, features or variations may be provided in addition to those set forth herein. For example, embodiments may be directed to various feature combinations and sub-combinations described in the detailed description.

### BRIEF DESCRIPTION OF THE DRAWINGS

[005] The accompanying drawings, which are incorporated in and constitute a part of  
30 this disclosure, illustrate various embodiments of the present invention. In the drawings:

[006] FIG. 1 is a block diagram of an operating environment;

[007] FIG. 2 is a flow chart of a method for providing a platform extensibility framework; and

[008] FIG. 3 is a block diagram of a system including a computing device.

#### DETAILED DESCRIPTION

[009] The following detailed description refers to the accompanying drawings.

Wherever possible, the same reference numbers are used in the drawings and the  
5 following description to refer to the same or similar elements. While embodiments of the invention may be described, modifications, adaptations, and other implementations are possible. For example, substitutions, additions, or modifications may be made to the elements illustrated in the drawings, and the methods described herein may be modified by substituting, reordering, or adding stages to the disclosed methods. Accordingly, the  
10 following detailed description does not limit the invention. Instead, the proper scope of the invention is defined by the appended claims.

[010] A platform extensibility framework may be provided. Consistent with embodiments of the present invention, a visualization platform may provide an intermediary between user applications and add-ons that provide extended features. Add-  
15 ons may be developed by third parties and used by any application comprising support for the visualization platform without needing to wait for a formal product release.

[011] FIG. 1 is a block diagram of an operating environment 100 for providing a visualization platform 110. Visualization platform 110 may comprise an object manager 115 and a layout engine 120. Visualization platform 110 may comprise an architecture  
20 allowing the creation of a data visualization, such as a chart, and may expose an interactive feature on the visualization. The architecture may provide integration with multiple rendering platforms. When a user selects the exposed feature, the architecture may translate the selection into a common format and modify the data visualization according to layout rules independent of the rendering platform.

[012] Operating environment 100 may further comprise a feature extension 130, an application 140 comprising a rendering component 145, and an output device 150. Feature extension 130 may comprise, for example, a functionality add-on providing a drawing instruction override, additional data, and/or additional drawing elements. Application 140  
25 may comprise, for example, a user application such as a word processing application, a spreadsheet application, a web browser, an email application, and/or other software application. Output device 150 may comprise, for example, a screen, a printer, an  
30 electronic document, and/or a web page.



[013] Visualization platform 110 may comprise a shared core comprising software libraries and/or utilities for providing interactive visualizations. The shared core may be implemented, for example, in C++ or C#, and may be platform independent. The shared core may comprise visualization utilities for providing layouts, shapes and/or geometry, line services, 3-dimensional rendering, animation frame generation, and/or interactive hotspots. Visualization platform 110 may further comprise application programming interfaces (APIs) for interacting with application 140 and/or feature extension 130.

[014] Application 140 and feature extension 130 may interface with visualization platform 110 to create objects in memory that may be rendered to output device 150 by application 140. Feature extension 130 may interface with visualization platform 130 rather than directly with application 140. Application 140 and feature extension 130 may be allowed to create the object together in a cooperative and independent manner. For example, application 140 may create a chart object with associated data bindings and feature extension 130 may create a legend and a series of data objects. Consistent with embodiments of the invention, feature extension 130 may create the object but leave it up to application 140 to provide the data.

[015] Operating system independence by operating environment 100 may be supported by a plurality of versions of visualization platform 110. Visualization platform 110 may be implemented using a platform-independent code, such as in a dynamically linked library (DLL) implemented using the .NET protocols developed by Microsoft Corporation of Redmond, WA. Thus, feature extension 130 may comprise an add-on that may be operative on any operating system supporting the platform-independent code. Consistent with embodiments of the invention, applications may not support the platform-independent code but may support the use of platform-independent feature extension 130 through the use of an operating system specific version of visualization platform 110.

[016] FIG. 2 is a flow chart setting forth the general stages involved in a method 200 consistent with an embodiment of the invention for providing a platform extensibility framework. Method 200 may be implemented using a computing device 300 as described in more detail below with respect to FIG. 3. Ways to implement the stages of method 200 will be described in greater detail below. Method 200 may begin at starting block 205 and proceed to stage 210 where computing device 300 may instantiate a visualization platform. For example, computing device 300 may instantiate visualization platform 110

comprising a single code base operative to run in multiple environments and/or operating systems.

[017] From stage 210, where computing device 300 instantiated, method 200 may advance to stage 215 where computing device 300 may register at least one feature extension with the visualization platform. For example, computing device 300 may locate feature extension 130 stored in a common memory location designated for feature extensions and feature extension 130 may provide details regarding available functionality to visualization platform 110. Consistent with embodiments of the invention, visualization platform 110 may determine and provide a list of objects that visualization platform 110 is operative to create. The list of createable objects may comprise objects for which drawing instructions are available from at least one of visualization platform 110, feature extension 130, and/or application 140.

[018] Once computing device 300 registers the at least one feature extension in stage 215, method 200 may continue to stage 220 where computing device 300 may receive a request to create an object. For example, visualization platform 110 may receive a request from application 140 to create an object, such as a chart, a graph, a picture, and/or a multimedia object. The request may comprise an object type, a data structure, at least one drawing instruction, a data series, and/or at least one data value.

[019] After computing device 300 receives the request to create the object in stage 220, method 200 may proceed to stage 225 where computing device 300 may create the object. For example, object manager 115 of visualization platform 110 may create the object according to at least one default drawing instruction associated with visualization platform 110 for an object of a requested type.

[020] From stage 225, method 200 may proceed to stage 230 where computing device 300 may determine whether the requested object is associated with at least one feature extension. For example, computing device 300 may determine whether the creation and/or modification of the requested object may make use of a drawing instruction, a data value, and/or another provided functionality that may be provided by feature extension 130. If the requested object is determined not to be associated with the at least one feature extension, method 200 may advance to stage 250 where computing device 300 may provide access to the created object to the requesting application as described in greater detail below.



[021] If, at stage 230, computing device 300 determines that the requested object is associated with the at least one feature extension, method 200 may advance to stage 235 where computing device 300 may determine whether the at least one feature extension is compatible with the requestor. For example, application 140 may request the creation of a chart object by visualization platform 110 that may request use of a data series provided by feature extension 130. Visualization platform 110 may verify the compatibility between the requested chart object and the requested data series. Consistent with embodiments of the invention, verifying compatibility may comprise validating that the requested data series is designated for use with a chart object of the requested type.

Further consistent with embodiments of the invention, visualization platform 100 may verify that application 140 comprises a compatible version with feature extension 130 and/or visualization platform 110. For example, visualization platform 110 may verify that application 140 comprises a version release number (e.g. 2.0), of a designated value or higher, and/or that application 140 comprises a particular release tier (e.g. Standard, Professional, Home, Business, and/or Educational). Visualization platform 110 may also verify whether application 140 comprises a valid license to make use of feature extension 130.

[022] If, at stage 235, computing device 300 determines that the at least one feature extension is not compatible with the requesting application, method 200 may advance to stage 250, described below. If computing device 300 determines that the at least one feature extension is compatible with the requesting application, method 200 may advance to stage 240 where computing device 300 may receive at least one object modification from the at least one feature extension. Consistent with embodiments of the invention, feature extension 130 may add, remove, and/or override at least one drawing instruction provided by visualization platform 110 and/or application 140 for the creation of the requested object. For example, feature extension 130 may provide a legend, a background, and/or an additional data value and/or data series to be added to a chart object. Further consistent with embodiments of the invention, feature extension 130 may provide drawing instructions comprising alternate shape geometry for data points on a chart object that override drawing instructions for the shape geometry as provided by application. For example, feature extension 130 may provide drawing instructions comprising an octagon shape for use as a data point on a chart object instead of a default circular shape provided by application 140. As another example, feature extension 130 may provide overriding

drawing instructions for modifying the object such as by re-drawing a chart as a log scale rather than a linear scale and/or using a split axis.

[023] From stage 240, where computing device 300 received at least one object modification, method 200 may advance to stage 245 where computing device 300 may  
5 modify the object in accordance with the received modification. For example, visualization platform 110 may draw a new element or redraw an already drawn element of the created object. Consistent with embodiments of the invention, visualization platform 110 may receive modifications from feature extension 130 and/or application 140 and may modify the object accordingly. For example, visualization platform 110 may  
10 create a chart object with a displayed data value according to a request from application 140, add a data series according to a modification received from feature extension 130, and modify a color of the data series displayed on the object according to a modification received from application 140.

[024] From stage 245, method 200 may advance to stage 250 where computing device  
15 300 may provide access to the created object to the requesting application. For example, visualization platform 110 may create the object in a memory location associated with the visualization platform and grant access to the memory location to feature extension 130 and/or application 140 using an application programming interface (API). Consistent with  
20 embodiments of the invention, visualization platform 110 may provide a copy of the object to the requestor rather than providing access to the object in the memory location associated with visualization platform 110.

[025] From stage 250, method 200 may advance to stage 255 where computing device  
300 may display the created object. For example, application 140 may access the created object and render it to display device 150 using rendering component 145. Rendering the  
25 object may comprise generating an image from a model associated with the object. The model associated with the object may comprise a description of three-dimensional objects in a data structure and may comprise geometry, viewpoint, texture, lighting, and/or shading information. The image may comprise a digital image or raster graphics image.

[026] Once computing device 300 displays the object in stage 255, method 200 may then  
30 end at stage 260.

[027] As an example of method 200, application 140 may request a chart object from visualization platform 110, wherein the request provides a chart type, at least one data value and/or data series. Visualization platform 110 may create the chart object and



populate the chart object with a legend and a title according to drawing instructions associated with visualization platform 110. At stage 255, where application 140 renders the chart object, application 140 may ask visualization platform 110 to provide the shapes associated with the chart object. Application 140 may walk through the shapes and  
5 receive drawing instructions for each in order to render them to an output device such as a printer or display device 150. Feature extension 130 may override and/or extend all kinds of visual elements of the created object. For example, they may add a new series type such as a line series, a column, and/or a pie series. For another example, feature extension 130 may override a particular map shape, a legend, a title, a legend item, an axis, and/or  
10 the entire chart.

[028] Each element and/or function provided by feature extension 130 may be associated with an element name, and those element names may be provided as part of providing the list of objects that visualization platform 110 may be operative to create. Visualization platform 110 may determine that the requested object is associated with feature extension  
15 130 by including the element names associated with feature extension 130 in the object creation request. For example, application 140 may provide a drop-down of available charts and/or data series selectable by the user. Upon receiving the user's selection, application 140 may request the object be created by visualization platform 110, and visualization platform 110 may query each known feature extension such as feature  
20 extension 130 to determine whether the feature extension is associated with the named element. Once visualization platform 110 determines the applicable feature extension, visualization platform 110 may receive drawing instructions from the applicable feature extension for adding, updating, and/or modifying the named element.

[029] An embodiment consistent with the invention may comprise a system for providing  
25 an extensibility framework. The system may comprise a memory storage and a processing unit coupled to the memory storage. The processing unit may be operative to instantiate a visualization platform, register at least one feature extension with the visualization platform, receive a request to create at least one object from at least one application, and create the at least one object. determining whether the request is associated with the at  
30 least one feature extension; in response to determining that the request is associated with the at least one feature extension: receiving at least one object modification from the at least one feature extension, and modifying the object according to the at least one object modification; and rendering the object on a display device.

[030] Another embodiment consistent with the invention may comprise a system for providing application extensibility. The system may comprise a memory storage and a processing unit coupled to the memory storage. The processing unit may be operative to locate at least one feature extension operative for use with a visualization platform,  
5 receive a request to create at least one object from at least one user application, and create the at least one object. The system may determine whether the request is associated with the at least one feature extension, and in response to determining that the request is associated with the at least one feature extension, receive at least one object modification from the at least one feature extension and modify the object according to the at least one  
10 object modification. The system may then render the object on a display device.

[031] Yet another embodiment consistent with the invention may comprise a system for providing an application extensibility framework. The system may comprise a memory storage and a processing unit coupled to the memory storage. The processing unit may be operative to instantiate a visualization platform, register at least one feature extension,  
15 provide a list of objects that the visualization platform is operative to create to an application, receive a request to create at least one of the list of objects from the application, and create the at least one requested object according to at least one drawing instruction provided by the visualization platform and associated with the requested object type. If the object is associated with a feature extension, and the feature extension is  
20 compatible with the at least one application, the system may be operative to receive at least one object modification from the at least one feature extension, modify the object according to the at least one object modification, and render the object on a display device.

[032] FIG. 3 is a block diagram of a system including computing device 300. Consistent with an embodiment of the invention, the aforementioned memory storage and processing  
25 unit may be implemented in a computing device, such as computing device 300 of FIG. 3. Any suitable combination of hardware, software, or firmware may be used to implement the memory storage and processing unit. For example, the memory storage and processing unit may be implemented with computing device 300 or any of other computing devices 318, in combination with computing device 300. The aforementioned  
30 system, device, and processors are examples and other systems, devices, and processors may comprise the aforementioned memory storage and processing unit, consistent with embodiments of the invention. Furthermore, computing device 300 may comprise an



operating environment for system 100 as described above. System 100 may operate in other environments and is not limited to computing device 300.

[033] With reference to FIG. 3, a system consistent with an embodiment of the invention may include a computing device, such as computing device 300. In a basic configuration, computing device 300 may include at least one processing unit 302 and a system memory 304. Depending on the configuration and type of computing device, system memory 304 may comprise, but is not limited to, volatile (e.g. random access memory (RAM)), non-volatile (e.g., read-only memory (ROM)), flash memory, or any combination. System memory 304 may include operating system 305, one or more programming modules 306, and may include visualization platform 110 and/or feature extension 130. Operating system 305, for example, may be suitable for controlling computing device 300's operation. In one embodiment, programming modules 306 may include user application 140. Furthermore, embodiments of the invention may be practiced in conjunction with a graphics library, other operating systems, or any other application program and is not limited to any particular application or system. This basic configuration is illustrated in FIG. 3 by those components within a dashed line 308.

[034] Computing device 300 may have additional features or functionality. For example, computing device 300 may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in FIG. 3 by a removable storage 309 and a non-removable storage 310. Computer storage media may include 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. System memory 304, removable storage 309, and non-removable storage 310 are all computer storage media examples (i.e., memory storage). Computer storage media may include, but is not limited to, RAM, ROM, electrically erasable read-only memory (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store information and which can be accessed by computing device 300. Any such computer storage media may be part of device 300. Computing device 300 may also have input device(s) 312 such as a keyboard, a mouse, a pen, a sound input device, a touch input

device, etc. Output device(s) 314 such as a display, speakers, a printer, etc. may also be included. The aforementioned devices are examples and others may be used.

[035] Computing device 300 may also contain a communication connection 316 that may allow device 300 to communicate with other computing devices 318, such as over a  
5 network in a distributed computing environment, for example, an intranet or the Internet. Communication connection 316 is one example of communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The  
10 term “modulated data signal” may describe a signal that has one or more characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media. The term computer readable media as  
15 used herein may include both storage media and communication media.

[036] As stated above, a number of program modules and data files may be stored in system memory 304, including operating system 305. While executing on processing unit 302, programming modules 306 (e.g. visualization platform 110 and/or feature extension 130) may perform processes including, for example, one or more of method 200's stages  
20 as described above. The aforementioned process is an example, and processing unit 302 may perform other processes. Other programming modules that may be used in accordance with embodiments of the present invention may include electronic mail and contacts applications, word processing applications, spreadsheet applications, database applications, slide presentation applications, drawing or computer-aided application  
25 programs, etc.

[037] Generally, consistent with embodiments of the invention, program modules may include routines, programs, components, data structures, and other types of structures that may perform particular tasks or that may implement particular abstract data types. Moreover, embodiments of the invention may be practiced with other computer system  
30 configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. Embodiments of the invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked



through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[038] Furthermore, embodiments of the invention may be practiced in an electrical circuit comprising discrete electronic elements, packaged or integrated electronic chips  
5 containing logic gates, a circuit utilizing a microprocessor, or on a single chip containing electronic elements or microprocessors. Embodiments of the invention may also be practiced using other technologies capable of performing logical operations such as, for example, AND, OR, and NOT, including but not limited to mechanical, optical, fluidic, and quantum technologies. In addition, embodiments of the invention may be practiced  
10 within a general purpose computer or in any other circuits or systems.

[039] Embodiments of the invention, for example, may be implemented as a computer process (method), a computing system, or as an article of manufacture, such as a computer program product or computer readable media. The computer program product may be a computer storage media readable by a computer system and encoding a computer program  
15 of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process. Accordingly, the present invention may be embodied in hardware and/or in software (including firmware, resident software, micro-code, etc.). In other words, embodiments of the present invention  
20 may take the form of a computer program product on a computer-usable or computer-readable storage medium having computer-usable or computer-readable program code embodied in the medium for use by or in connection with an instruction execution system. A computer-usable or computer-readable medium may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with  
25 the instruction execution system, apparatus, or device.

[040] The computer-usable or computer-readable medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific computer-readable medium examples (a non-exhaustive list), the computer-readable medium may include the  
30 following: an electrical connection having one or more wires, a portable computer diskette, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, and a portable compact disc read-only memory (CD-ROM). Note that the computer-usable or

computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted, or otherwise processed in a suitable manner, if necessary, and then stored in a computer memory.

5 [041] Embodiments of the present invention, for example, are described above with reference to block diagrams and/or operational illustrations of methods, systems, and computer program products according to embodiments of the invention. The functions/acts noted in the blocks may occur out of the order as shown in any flowchart. For example, two blocks shown in succession may in fact be executed substantially  
10 concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved.

[042] While certain embodiments of the invention have been described, other embodiments may exist. Furthermore, although embodiments of the present invention have been described as being associated with data stored in memory and other storage  
15 mediums, data can also be stored on or read from other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or a CD-ROM, a carrier wave from the Internet, or other forms of RAM or ROM. Further, the disclosed methods' stages may be modified in any manner, including by reordering stages and/or inserting or deleting stages, without departing from the invention.

20 [043] All rights including copyrights in the code included herein are vested in and the property of the Applicant. The Applicant retains and reserves all rights in the code included herein, and grants permission to reproduce the material only in connection with reproduction of the granted patent and for no other purpose.

[044] While the specification includes examples, the invention's scope is indicated by  
25 the following claims. Furthermore, while the specification has been described in language specific to structural features and/or methodological acts, the claims are not limited to the features or acts described above. Rather, the specific features and acts described above are disclosed as example for embodiments of the invention.



WHAT IS CLAIMED IS:

1. A method (200) for providing an extensibility framework, the method (200) comprising:
  - instantiating (210) a visualization platform (110);
  - registering (215) at least one feature extension (130) with the visualization platform (110);
  - receiving (220) a request to create at least one object from at least one application (140);
  - creating (225) the at least one object;
  - determining (230) whether the request is associated with the at least one feature extension (130);
  - in response to determining (230) that the request is associated with the at least one feature extension (130):
    - receiving (240) at least one object modification from the at least one feature extension (130), and
    - modifying (245) the object according to the at least one object modification; and
    - rendering (255) the object on a display device (150).
2. The method (200) of Claim 1, wherein the at least one application (140) renders the object on the display device (150).
3. The method (200) of Claim 1, further comprising:
  - receiving at least one second object modification from the at least one application (140); and
  - modifying the object according to the at least one second object modification.
4. The method (200) of Claim 1, wherein receiving the request comprises receiving a visualization type and at least one data value;
  - wherein the object comprises a visualization object of the requested visualization type; and
  - wherein the visualization platform (110) comprises at least one default drawing instruction for the visualization type and the at least one data value.
5. The method (200) of Claim 1, wherein the application (140) is operative to determine a plurality of objects the visualization platform (110) is operative to create and display the plurality of objects to a user.

6. The method (200) of Claim 5, wherein determining the plurality of objects the visualization platform (110) is operative to create comprises retrieving a list of the plurality of objects from the visualization platform (110).
7. The method (200) of Claim 5, wherein determining the plurality of objects the visualization platform (110) is operative to create comprises searching at least one location for the at least one feature extension (130).
8. The method (200) of Claim 1, wherein determining (230) whether the request is associated with the at least one feature extension (130) comprises determining whether the at least one feature extension (130) comprises at least one of the following: at least one data value and at least one drawing instruction.
9. The method (200) of Claim 1, wherein the at least one object modification from the at least one feature extension (130) comprises a drawing instruction overriding at least one drawing instruction used by the visualization platform (110) to create the object.
10. The method (200) of Claim 1, wherein the at least one object modification from the at least one feature extension (130) comprises a drawing instruction adding at least one drawing instruction to be used by the visualization platform (110) to create the object.
11. The method (200) of Claim 1, wherein the at least one object modification from the at least one feature extension (130) comprises a plurality of data points to be displayed on the object.
12. A system (300) for providing application (140) extensibility, the system (300) comprising:
  - a memory storage (309, 310); and
  - a processing unit (302) coupled to the memory storage (309, 310), wherein the processing unit (302) is operative to:
    - instantiate (210) a visualization platform (110);
    - locate at least one feature extension (130) operative for use with the visualization platform (110);
    - receive (220) a request to create at least one object from at least one user application (140);
    - create (225) the at least one object;
    - determine (230) whether the request is associated with the at least one feature extension (130);



in response to determining that the request is associated with the at least one feature extension (130):

receive (240) at least one object modification from the at least one feature extension (130), and

modify (245) the object according to the at least one object modification; and

rendering (255) the object on a display device (150).

13. The system (300) of Claim 12, further comprising being operative to provide a list of objects the visualization platform (110) is operative to create to the at least one user application (140), wherein the list of objects the visualization platform (110) is operative to create comprises at least one object requiring the at least one object modification from the at least one feature extension (130).

14. The system (300) of Claim 13, wherein being operative to provide a list of objects the visualization platform (110) is operative to create to the at least one user application (140) comprises being operative to determine (235) whether the at least one user application (140) is compatible with the at least one feature extension (130); and

in response to determining (235) that the at least one user application (140) is not compatible with the at least one feature extension (130), withholding at least one object associated with the at least one feature extension (130) from the list of objects the visualization platform (110) is operative to create.

15. A computer-readable medium which stores a set of instructions which when executed performs a method (200) for providing an application (140) extensibility framework, the method (200) executed by the set of instructions comprising:

instantiating (210) a visualization platform (110);

registering (215) at least one feature extension (130) with the visualization platform (110);

providing a list of objects that the visualization platform (110) is operative to create to at least one application (140), wherein the list of objects comprises at least one object the visualization platform (110) requires the at least one feature extension (130) to create;

receiving (220) a request to create at least one of the list of objects from at least one application (140), wherein the request comprises a user selection from the list of objects provided to the application (140);

creating (225) the at least one requested object according to at least one drawing instruction provided by the visualization platform (110) and associated with the requested object type;

determining (230) whether the requested object is associated with the at least one feature extension (130);

in response to determining (230) that the request is associated with the at least one feature extension (130), determining (235) whether the at least one feature extension (130) is compatible with the at least one application (140), wherein determining (235) whether the at least one feature extension (130) is compatible with the at least one application (140) comprises at least one of the following: determining whether the at least one application (140) comprises a compatible release version and determining whether the at least one application (140) comprises a license to use the at least one feature extension (130);

in response to determining (235) that the at least one feature extension (130) is compatible with the at least one application (140):

receiving (240) at least one object modification from the at least one feature extension (130), wherein the at least one object modification comprises at least one of the following: a drawing instruction override and at least one additional drawing element, and

modifying (245) the object according to the at least one object modification; and

rendering (255) the object on a display device (150) by the application (140).



1/3

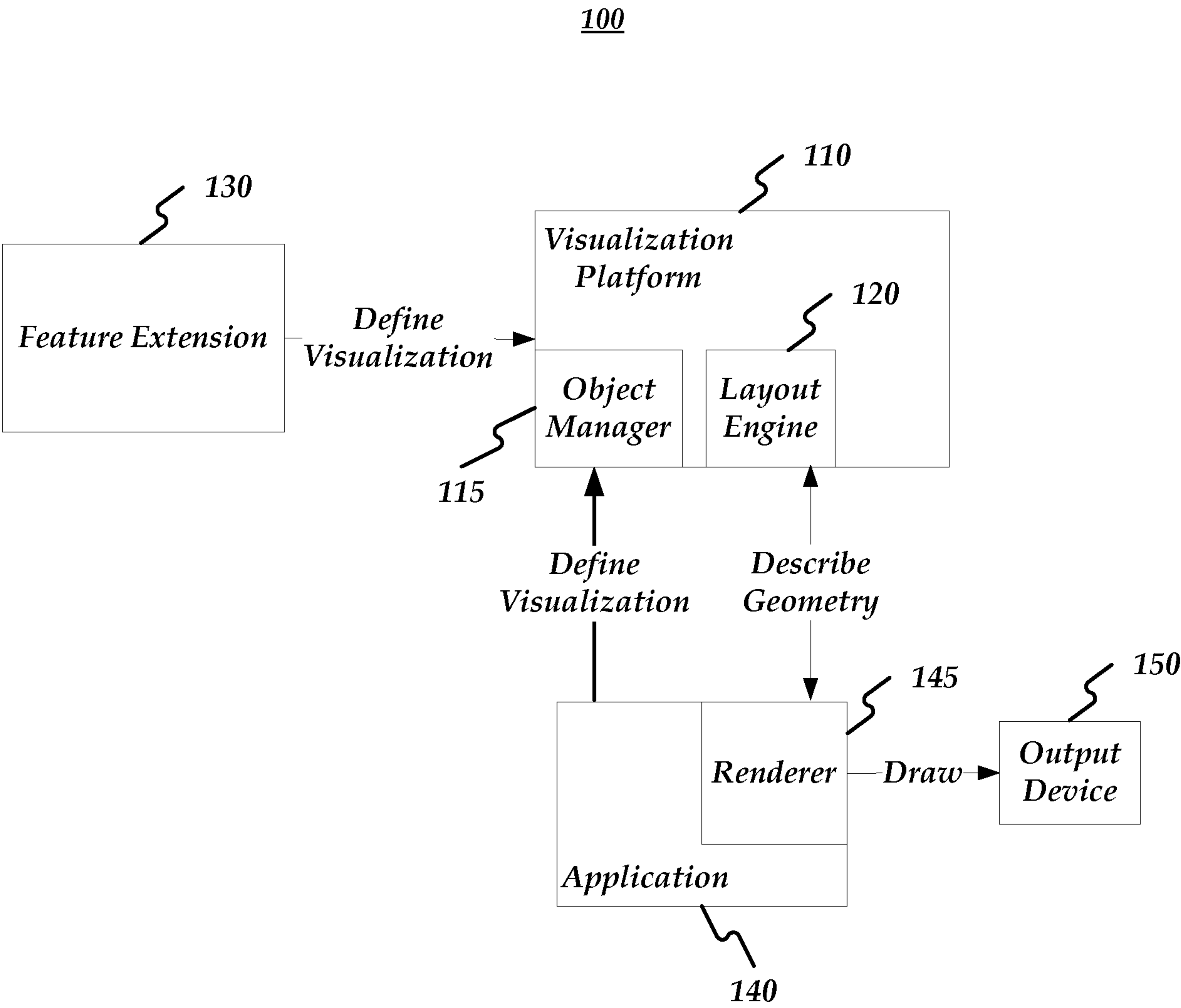


FIG. 1

200

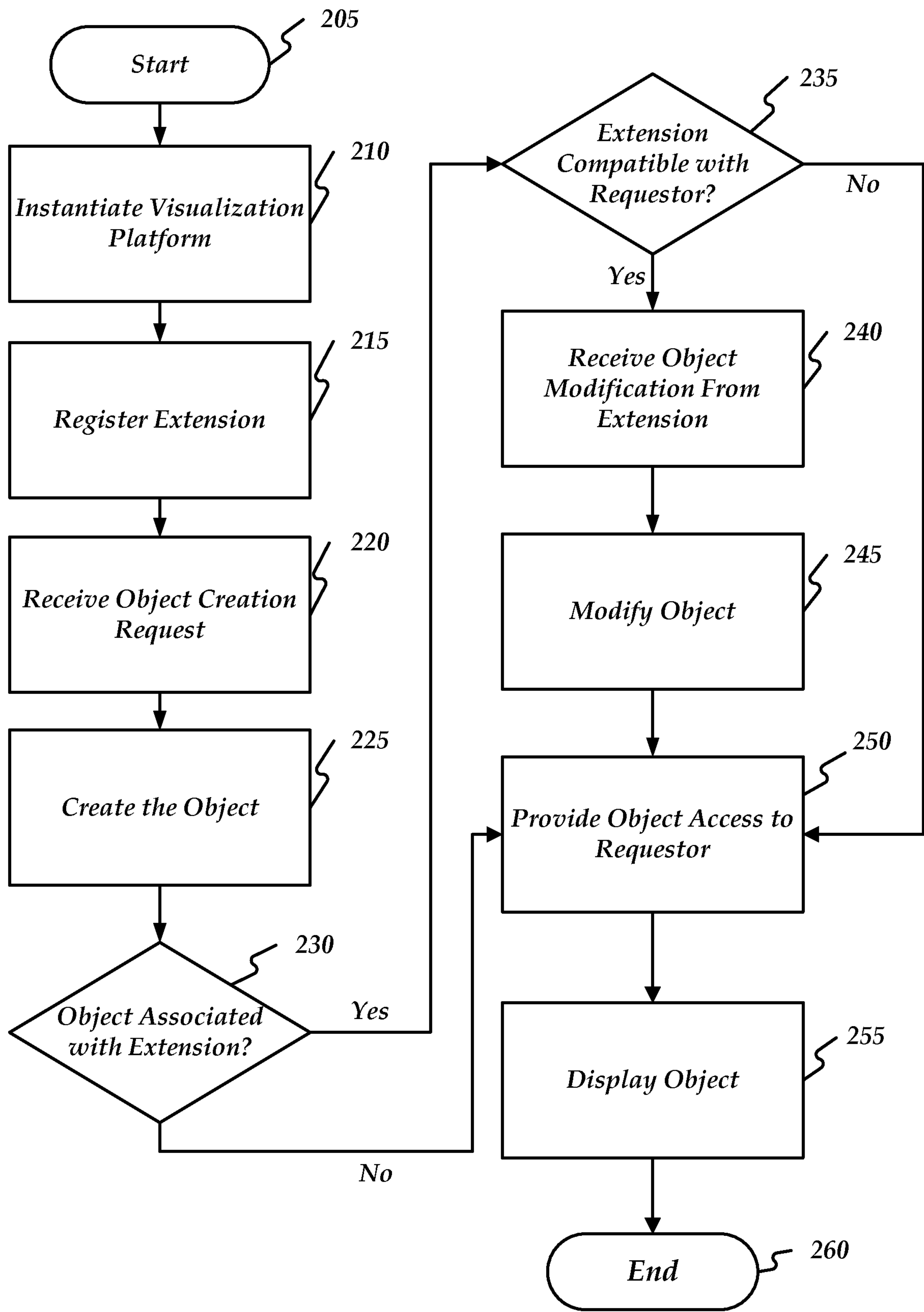


FIG. 2



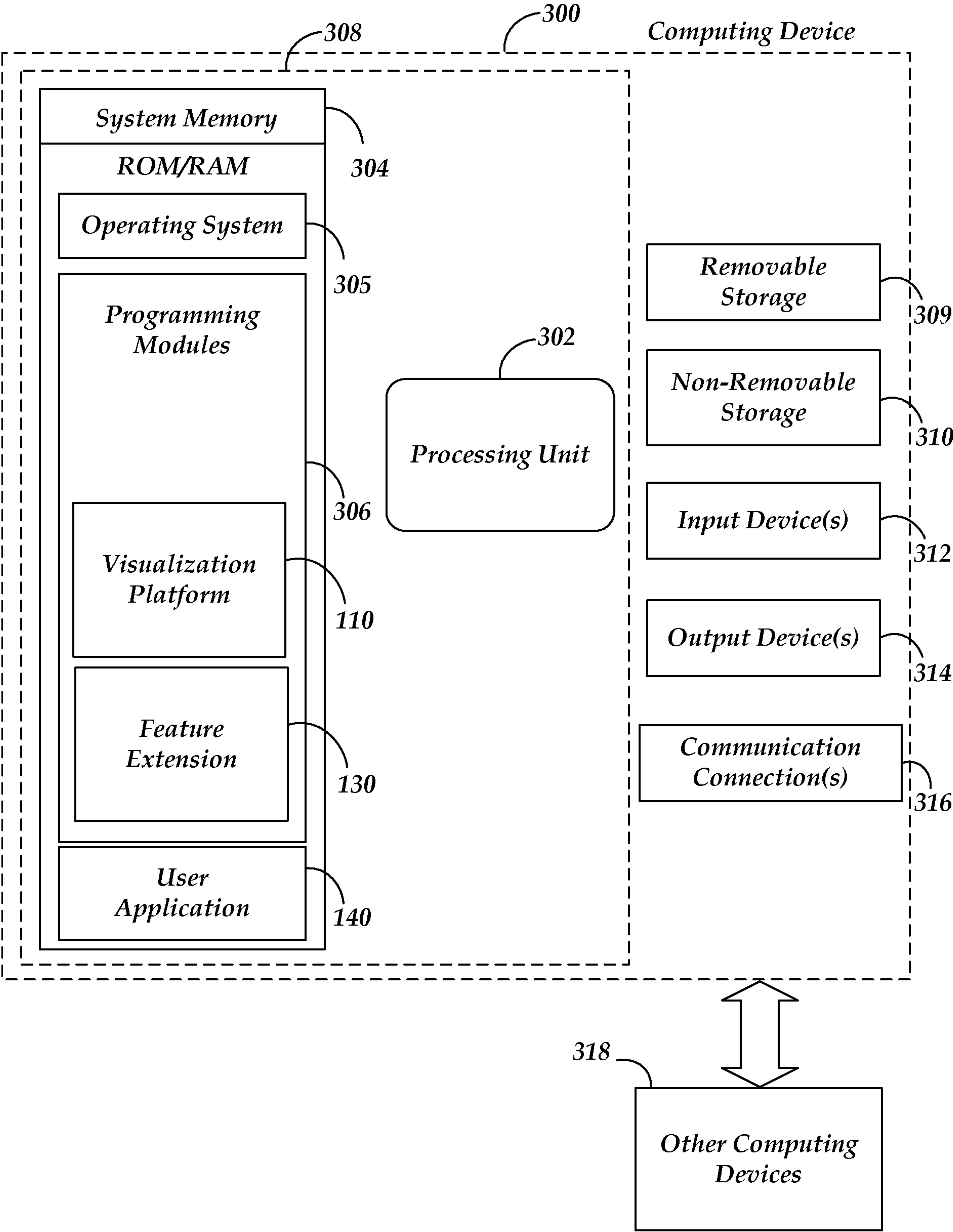
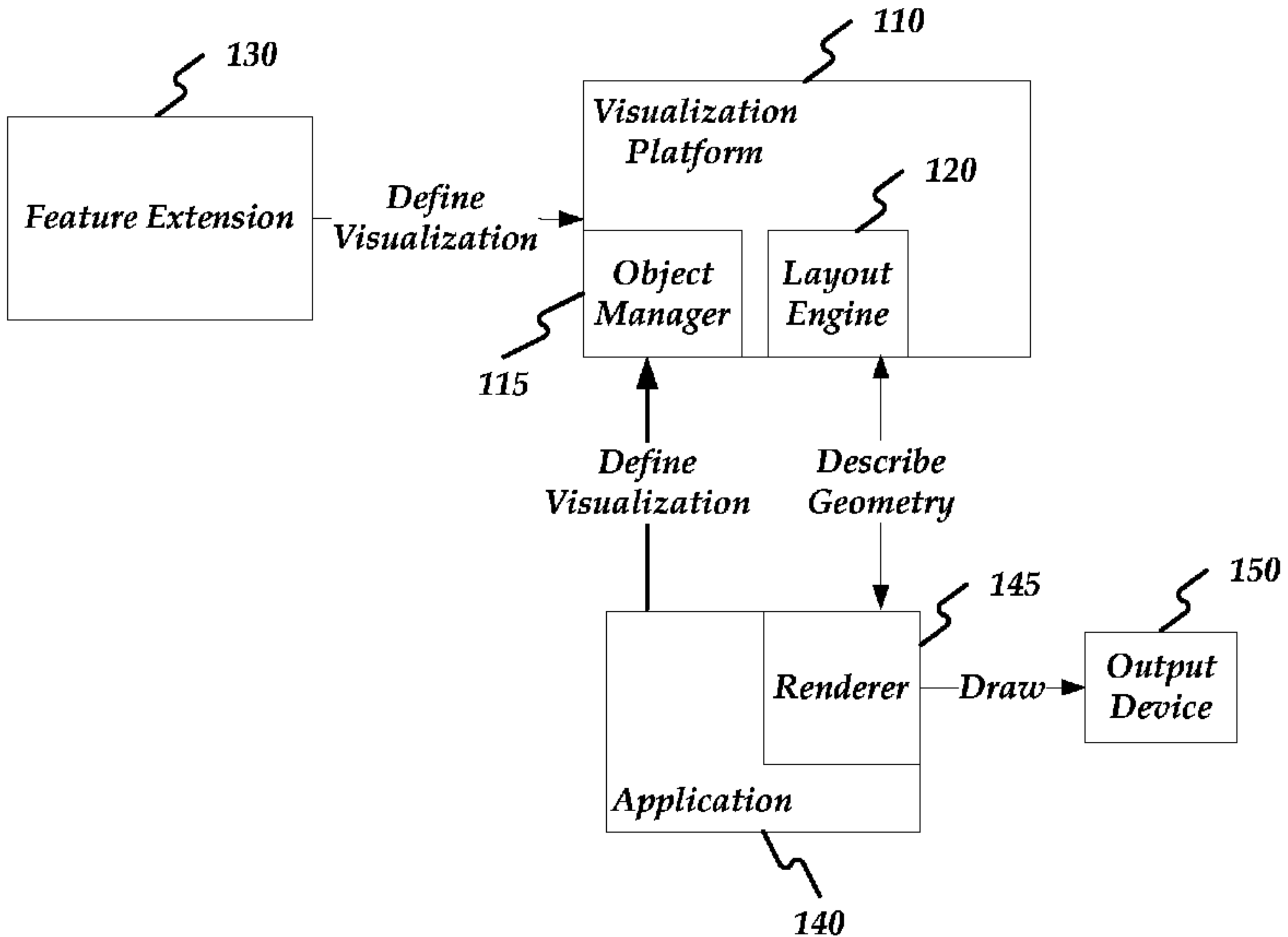


FIG. 3

100



**FIG. 1**