



- (51) International Patent Classification: Not classified
- (21) International Application Number: PCT/US2014/043747
- (22) International Filing Date: 24 June 2014 (24.06.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:

61/840,247	27 June 2013 (27.06.2013)	US
14/157,756	17 January 2014 (17.01.2014)	US
- (71) Applicant: MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) Inventors: KUMAR, Deepak; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). DESAI, Aditya Makarand; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). BASU, Arijit; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). YU, Yen Ping; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). SWAMINATHAN, Manoj; c/o Microsoft Corporation, LCA - International Patents (8/1172), 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, BN, 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, IR, IS, JP, KE, KG, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, 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:

- without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: AUTOMATIC CONFIGURATION OF A COMPUTER SYSTEM BASED ON PROCESS MODELING OF AN IMPLEMENTED PROCESS

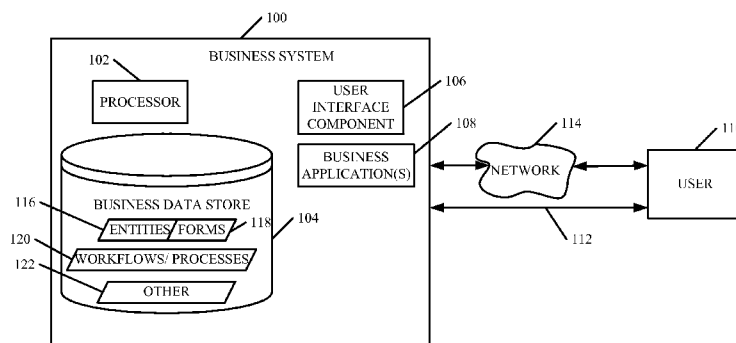


FIG. 1

(57) Abstract: A configuration architecture automatically links the output of a business process modeler to a business system configuration tool. The output of the business process modeler is translated into a representation of a source business process that is used by the business system configuration tool to automatically generate configuration templates. A user can provide configuration information through the configuration templates. The configuration inputs are applied to an implementation of a business system to obtain a configured, working business system instance with the same functionality as the source business process.

WO 2014/209922 A2

AUTOMATIC CONFIGURATION OF A COMPUTER SYSTEM BASED ON PROCESS MODELING OF AN IMPLEMENTED PROCESS

BACKGROUND

5 [0001] Computer systems are currently in wide use. In order to implement an instance of a computer system, the computer system is often modeled first and then configured.

[0002] By way of example, some computer systems include business systems, such as enterprise resource planning (ERP) systems, customer resource management (CRM) systems, line-of-business (LOB) systems, etc. In order to implement an instance of a
10 business system, the business system is first often modeled by one or more analysts. The system is then often configured by one or more consultants. The analysts and consultants are often on two separate teams and information is often lost when the process moves from the analysis (or modeling) part of the process to the configuration (or implementation) part of the process. Also, this type of process has conventionally been almost entirely manual.

15 [0003] More specifically, the team of analysts often analyzes and models the business system before any configuration is done. The configuration consultant team then obtains information that represents the analysis and modeling, and configures a system based on that information. However, it is common for at least some of the modeling information generated by the team of analysts to be lost (or otherwise not carried forward) when the
20 information is transferred for configuration.

[0004] Therefore, once the team of consultants configures the business system, it often does not entirely match the model that the analysts had created. This results in a relatively large amount of rework that is needed in order to make the configured system match the modeled system. Thus, because all of this is almost entirely manual, return on investment
25 realized for the business system suffers.

[0005] The discussion above is merely provided for general background information and is not intended to be used as an aid in determining the scope of the claimed subject matter.

SUMMARY

[0006] A configuration architecture automatically links the output of a business process
30 modeler to a business system configuration tool. The output of the business process modeler is translated into a representation of a source business process that is used by the business system configuration tool to automatically generate configuration templates. A user can provide configuration information through the configuration templates. The configuration inputs are applied to an implementation of a business system to obtain a

configured, working business system instance with the same functionality as the source business process.

[0007] This Summary is provided to introduce a selection of concepts in a simplified form that are 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 as an aid in determining the scope of the claimed subject matter. The claimed subject matter is not limited to implementations that solve any or all disadvantages noted in the background.

BRIEF DESCRIPTION OF THE DRAWINGS

- 10 [0008] FIG. 1 is a block diagram of one illustrative business system.
- [0009] FIG. 2 is a block diagram of one illustrative embodiment of a configuration architecture.
- [0010] FIG. 3 is a flow diagram illustrating one embodiment of the operation of the configuration architecture shown in FIG. 2.
- 15 [0011] FIGS. 3A and 3B are illustrative user interface displays.
- [0012] FIG. 3C shows one illustrative mapping between an output of a business process modeler and a business system configuration tool.
- [0013] FIG. 3D shows one embodiment of a template and question generation model.
- [0014] FIGS. 3E-3I are illustrative user interface displays.
- 20 [0015] FIG. 4 shows one embodiment of the configuration architecture shown in FIG. 2 deployed in various other architectures.
- [0016] FIGS. 5-10 show various embodiments of mobile devices.
- [0017] FIG. 11 is a block diagram of one illustrative computing environment.

DETAILED DESCRIPTION

- 25 [0018] FIG. 1 shows a block diagram of one embodiment of a business system 100. Business system 100 includes processor 102, business data store 104, user interface component 106 and one or more business applications 108. FIG. 1 shows that business system 100 can be accessed by user 110 either directly (as indicated by arrow 112) or over a network 114.
- 30 [0019] Business data store 104 illustratively includes a variety of different business information and business records. For instance, in one embodiment, business data store 104 includes entities 116, forms 118, workflows/processes 120, and other information 122. Entities 116 are illustratively business records that identify business items. For example, a vendor entity identifies and represents a vendor; a product entity identifies and represents

a product; a customer entity identifies and represents a customer; an opportunity entity identifies and represents a business opportunity, etc. Business applications 108 illustratively run workflows/processes 120 and operate on entities and other business data in business data store 104 to implement an instance of a business system. By way of
5 example, it can be an instance of an ERP system, a CRM system, an LOB system, etc. However, for the sake of simplicity, it will be described herein as an instance of an ERP system. Forms 118 illustratively present user interface elements to user 110 so that the user can access and manipulate business applications 108 and other items in business system 100.

10 **[0020]** FIG. 1 also shows that business data store 104 is a single data store and that it is local to business system 100. However, it can be multiple data stores and all can be local to business system 100, or remote therefrom or some can be local while others can be remote.

[0021] The business applications 108 illustratively include components or other types of
15 applications that run on business system 100 to perform business operations, tasks, activities, workflows, etc. For instance, a business application 108 can be a general ledger application, an inventory application, a billing application, among others.

[0022] Processor 102 is illustratively a computer processor with associated memory and timing circuitry (not separately shown). It is illustratively a functional part of business
20 system 100 and is activated by, and facilitates the functionality of, other applications and components or other items of business system 100.

[0023] User interface component 106 illustratively generates user interface displays, with corresponding user input mechanisms, for interaction by user 110. The user input mechanisms can take a wide variety of forms. For instance, they can be dropdown menus,
25 icons, links, check boxes, text boxes, buttons, etc. In addition, the user input mechanisms can be actuated in a wide variety of different ways. Where the display device has a touch sensitive screen, the user input mechanism can be actuated using touch gestures. In addition, where the device used by user 110 or business system 100 includes a speech recognition system, then the user input mechanisms can be activated by voice commands.
30 Also, the user input mechanisms can illustratively be actuated using a point and click device (such as a mouse or track ball), a keyboard, a keypad, a soft keyboard, a stylus, a thumbpad, or other methods.

[0024] In one embodiment, user 110 accesses business system 100 in order to perform business tasks, activities or operations or to generally manipulate business system 100 in

order to conduct business. User 110 illustratively accesses business system 100 by logging onto it, and providing authentication information (such as a user name and password). Of course, other ways of accessing business system instance 100 can be used as well.

[0025] FIG. 2 is a block diagram of one embodiment of a configuration architecture 130. FIG. 2 shows configuration architecture 130 coupled to business system 100 and user device 132. In general, configuration architecture 130 obtains a representation of a source business process from business system 100 (such as by having a user record one, as described below, or otherwise) and links it to business system configuration tool 142 to obtain configuration templates that can be used by user 134 to configure a business system with the same functionality as the source business process obtained from business system 100.

[0026] More specifically, configuration architecture 130 illustratively includes business process recorder 136, business process modeler 138, translator/synchronizer component 140, business system configuration tool 142 (that includes a template and question generation model 144), processor 146 and user interface component 148.

[0027] FIG. 2 also shows that user device 132 includes processor 150 and user interface component 142. User device 132 is shown generating user interface displays 154 with user input mechanisms 156 for interaction by user 134. User device 132 is also shown providing an output to an implementation of a business system instance, as indicated by block 158.

[0028] FIG. 3 is a flow diagram illustrating one embodiment of the overall operation of configuration architecture 130 in generating output 158. UI component 148 in configuration architecture 130 (or UI component 152 in user device 132) illustratively first provides a user interface display to user 134 so that configuration architecture 130 can access a recorded source business process that will include both a representation of the process, and configuration information related to the process. This is indicated by block 162 in FIG. 3. In one embodiment, user 134 operates business process recorder 136 to perform a business process within business system 100 so that business process recorder 136 can record the business process. In such an embodiment, business process recorder 136 provides a recorded business process 160 to business process modeler 138. Receiving user inputs and recording the source process is indicated by block 164.

[0029] In another embodiment, user 134 (or another user) has already recorded a business process and stored it in a place that is accessible by business process modeler 138. In that embodiment, business process modeler 138 simply accesses the already-

recorded business process 160. Accessing an already-recorded, stored source process is indicated by block 166. Of course, business process modeler 138 can obtain access to a recorded source business process in other ways as well, and this is indicated by block 168.

[0030] FIG. 3A shows one embodiment of a user interface display 200. User interface display 200 illustratively includes a hierarchical process structure 202 that has a plurality of nodes. The nodes that are indented with respect to other nodes are descendants (or child nodes) of the other nodes. Therefore, a node that is outdented with respect to an indented node is an ancestor or parent with respect to the immediately following indented nodes. In one embodiment, the processes represented by each node are also color coded to indicate where they fall in the hierarchy. User 134 can simply select one of the business processes by selecting one of the nodes in the hierarchy 202 in FIG. 3A. If an already-existing recording has been made for the corresponding process, that is provided to business process modeler 138 for modeling. If not, then user 134 is walked by business process recorder 136 through a series of user interface displays that allow user 134 to make a recording of the selected business process.

[0031] Once the user has selected the business process to be configured, user interface component 148 illustratively generates a user interface display with a user input mechanism that receives a user synchronization input to send the selected business process to a configuration tool. In response, user interface component 148 generates a user interface display such as display 204, shown in FIG. 3B. This includes a configuration system selection mechanism 206. In the embodiment shown in FIG. 3B, mechanism 206 is a dropdown menu, but it could be other mechanisms as well. The various instances of the configuration tool that the user has access to can be displayed in selectable form. Other instances can be displayed as well. In any case, user 134 illustratively selects a specific service instance of a configuration system that is to be used in configuring the selected process. Thus, the payload for the selected process will be delivered to the identified service instance selected by user 134.

[0032] In one embodiment, business process recorder 136 generates recorded business process 160 as a zip file which can be uploaded to a service instance of business process modeler 138. The zip file illustratively includes a variety of files (such as XML files) each corresponding to some activity (and hence each corresponding to a leaf node in the hierarchy 202). The files serve as the recorded business process 160 that is provided to business process modeler 138, and they illustratively include a description of tables that

are to be configured in order to complete the business process implementation for the selected process.

[0033] Having received its input, business process modeler 138 generates a process diagram (or another representation) for the selected business process. This is indicated by block 170 in the flow diagram of FIG. 3. The business process representation is indicated by block 172 in FIG. 2. Translator/synchronizer 140 then translates representation 172 for input to business configuration tool 142. The representation of the business process that is provided to business configuration tool 142 is indicated by block 174 in FIG. 2. Translating the representation 172 into the representation 174 is indicated by block 176 in the flow diagram of FIG. 3. This can be done, for instance, by providing representation 174 as an XML file (or files) with process information and other configuration information. This is indicated by block 178. Of course, representation 174 can take other forms as well, and this is indicated by block 179. Receiving the user selection input selecting a configuration tool instance is indicated by block 180 and sending the translated representation of the business process to the selected configuration tool instance 142 is indicated by block 181.

[0034] Table 1 shows one illustrative embodiment of a sample schema that embodies a representation 174 that is expected by business system configuration tool 142.

TABLE 1

```

20 <Framework>
    <Id>rrrr</Id>
    <Name>
    </Name>
    <Version>6.2:158</Version>
25 <SyncID>1BC10AFC-AEDA-4B67-8EC6-AC90FEAAA000</SyncID>
    <LeafID>DCEADEFc-6337-4E8B-86D1-6DCA9E214414</LeafID>
    <Level>
    <Category>aa</Category>
    <Name>aaaa</Name>
30 <Description>
    </Description>
    <OperationsGroup>None</OperationsGroup>
    <FolderName>aaaa_DCEADEFc-</FolderName>
    <HasRecording>True</HasRecording>

```

```

<ApplicationModule>Purchase order</ApplicationModule>
<RecordingInformation>
  <Tables>
    <Table>
5      <Name>BrazilParameters</Name>
      <TableGroup>Parameter</TableGroup>
      <CountryRegionCodes>BR</CountryRegionCodes>
      <AOSAuthorization>None</AOSAuthorization>
      <Label>Brazilian parameters</Label>
10     <ConfigurationKey>
      </ConfigurationKey>
      <DeveloperDocumentation>The BrazilParameters table contains parameters related
to Brazilian localization features</DeveloperDocumentation>
      <Shared>True</Shared>
15     <Fields>
      <Field>
      <Name>ChronologicalInvoiceNumbering</Name>
      <Label>Enable fix issue date</Label>
      <Type>Enum</Type>
20     <HelpText>Enable setting fixed issue dates at fiscal documents
types</HelpText>
      <IsRefRecId>False</IsRefRecId>
      <CountryRegionCodes>BR</CountryRegionCodes>
      </Field>
25    </Fields>
    </Table>
    <DMFEntities>
      <Table>
      <Name>Customer</Name>
30    </Table>
    </DMFEntities>
  </RecordingInformation>
  <SyncID>DCEADEFc-6337-4E8B-86D1-6DCA9E214414</SyncID>
</Level>

```

</Framework>

The class representation of this XML file is as below:

```
public class Framework : JsonEntityBase
5   {
    public string Id { get; set; }
    public string Name { get; set; }
    public string Version { get; set; }
    public Guid SyncId { get; set; }
10   public Guid LeafId { get; set; }
    public Node Level { get; set; }
  }

public class Node : JsonEntityBase
15   {
    public string Category { get; set; }
    public string Name { get; set; }
    public string Description { get; set; }
    public bool HasRecording { get; set; }
20   public Guid SyncId { get; set; }
    public Node Level { get; set; }
    public string AXApplicationVersion { get; set; }
    public string IndustryCode { get; set; }
    public string CountryCode { get; set; }
25   public Recording RecordingInformation { get; set; }
  }

public class Recording : JsonEntityBase
  {
30   public string AXApplicationVersion { get; set; }
    public string IndustryCode { get; set; }
    public List<Table> Tables;
    public List<Table> DMFEntities;
  }
```

```

public class Table : JsonEntityBase
{
    public string Name { get; set; }
5    public string Label { get; set; }
    public string DeveloperDocumentation { get; set; }
    public string TableGroup { get; set; }
    public string ConfigurationKey { get; set; }
    public string Shared { get; set; }
10    public string CountryRegionCodes { get; set; }
    public List<Field> Fields { get; set; }
}

public class Field : JsonEntityBase
15 {
    public string Name { get; set; }
    public string Label { get; set; }
    public string HelpText { get; set; }
    public string Type { get; set; }
20    public string CountryRegionCodes { get; set; }
}

```

[0035] Once the representation 174 is sent to business system configuration tool 142, template and question generation model 144 is used to generate and output configuration templates, and a template hierarchy, corresponding to the selected business process. This is indicated by block 182 in FIG. 3. Configuration templates are indicated at 184 in FIG. 2.

[0036] In one embodiment, the configuration templates include templates 186, subtemplates 188, functional areas 190, question groups 192 and questions 194. The configuration templates can also classify or group the questions that are applicable to various regions, industries, versions of the business system that generated the business process, etc. This is indicated by block 196 in FIG. 3. The configuration templates can, of course, include other information as well, as indicated by block 198.

[0037] The XML in Table 1 automatically generates a template hierarchy which shows a number of different things. First, it identifies the parent node of the business process which generated it. It also identifies the subnode for which the configuration templates are

generated, and all sub-templates and functional areas are based on a suitable heuristic. A list of questions and question groups are also automatically generated. In one embodiment, template and question generation model 144 reduces the number of questions to a minimal set and categorizes them for an easy answering experience. That is, it illustratively reduces the redundancy in the question and groups questions together so that the user can answer like questions in a group. This enhances the ability of the user to perform the process.

[0038] Once configuration templates 184 are generated, they can be provided to user 134 either directly, or through a user device such as user device 132. In the case where they are provided directly, UI component 148 generates user interface displays 154. However, where they are provided through user device 132, UI component 152 generates user input displays 154. Displays 154 present the templates, subtemplates, question groups and questions to user 134 so that user 134 can provide configuration inputs 250, through the templates, subtemplates, question groups and questions. Receiving user inputs through the configuration templates 184 is indicated by block 252 in the flow diagram of FIG. 3.

[0039] Table 2 below shows one illustrative description that template and question generation model 144 can use to automatically generate configuration templates 184. FIG. 3C shows one example of a mapping between the information 185 in Table 2 and the templates, functional areas and question groups 187 that are generated by business system configuration tool 142. It will be noted that the mapping in FIG. 3C is just one embodiment. Other mappings can be used as well. For instance, the information can be mapped to a functional area, or business module, instead of just to a business process. Also, the information in Table 2 and the map in FIG. 3 can be different. They can reflect any other type of hierarchy, or even a flat structure.

TABLE 2

Level 1 - Category	1.0 Develop Vision and Strategy (10002)
Represents the highest level of process in the enterprise, such as Manage customer service, Supply chain, Financial organization, and Human resources.	
Level 2 – Process Group	1.1 Define the business concept and long-term vision (10014)
Indicates the next level of processes and represents a group of processes. Perform after sales repairs, Procurement, Accounts payable, Recruit/source, and Develop sales strategy are examples of process groups.	
Level 3 – Process	1.1.1 Accesses the external environment

	(10017)
A series of interrelated activities that converts inputs into results (outputs); processes consume resources and require standards for repeatable performance; and processes respond to control systems that direct the quality, rate, and cost of performance.	
Level 4 – Activity	1.1.1.1 Analyze and evaluate competition (10021)
Indicates key events performed when executing a process. Examples of activities include Receive customer requests, Resolve customer complaints, and Negotiate purchasing contracts.	
Level 5 – Task	12.2.3.1.1 Identify project requirements and objectives (11117)
Tasks represent the next level of hierarchical decomposition after activities. Tasks are generally much more fine grained and may vary widely across industries. Examples include: Create business case and obtain funding and Design recognition and reward approaches.	

[0040] FIG. 3D is one embodiment of a high level view of template and question generation model 144.

[0041] FIG. 3E shows one embodiment of a set of templates 184 that can be generated. The set of templates are listed in a flat view in template pane 300. When the user selects a
 5 template from template pane 300, the template hierarchy 302 for the selected template is displayed. In addition, generation display 304 shows some items of detail regarding the generation of the template hierarchy 302, by business system configuration tool 142.

[0042] FIG. 3F shows a user interface display 306 that allows the user to select one of the templates in order to configure a business process or module corresponding to the
 10 template. The user can enter project information generally at 308, such as the project name, the method of configuration, the release version, the industry, the country or region, etc. The user can then select a template from template pane 310. The selected template, subtemplate and functional areas are then displayed in detail pane 312. The user can select one of the functional areas to review a set of questions or question groups that were
 15 automatically generated for that functional area.

[0043] User 134 can then provide configuration inputs 250 (shown in FIG. 2) through those questions in order to configure the business process.

[0044] FIGS. 3G-3I are illustrative user interface displays that show different question types, based upon the different metadata in the XML structure received from translator/synchronizer 140, at business system configuration tool 142. For instance, FIG. 3G shows that business system configuration tool 142 generates a table type question if the table to be configured is not a parameter table (as indicated by the metadata in Table 1).

[0045] FIG. 3H shows that business system configuration tool 142 generates a field type question if a field belonging to a parameter table is to be configured. In addition, FIG. 3I shows that business system configuration tool 142 generates a data import/export framework (DIEF) type question if a DIEF entity is to be configured.

[0046] It will be noted that there can be many other question types as well. For instance, the question types can include relations, abstract types, etc.

[0047] It should be noted that a large percent of parameters may be duplicated across different functional areas in a category. Therefore, in one embodiment, a functional area is created with a name referred to as "Parameters: Name of the category". This functional area contains all the parameters that are to be configured for the category. This avoids duplication and redundancy to expedite the process.

[0048] The questions and question groups, along with templates, subtemplates and functional areas are displayed for user 134 on UI displays 154. The user 134 provides configuration inputs 250 through the templates. This configuration information is then applied against an implementation of a business system to obtain a configured, working business system instance with the same functionality as the source business process. This is indicated by block 350 in FIG. 3.

[0049] It can thus be seen that configuration architecture 130 provides an automated way of creating configuration templates 184 and questions based on an input which represents a business process. Translator/synchronizer 140 automatically links business process modeler 138 to business system configuration tool 142 by providing the representation 174 of the business process being configured, in a form that is acceptable to business system configuration tool 142. The representation 174 can be an XML input file (or another file or it can be distributed in a database management system that the configuration architecture has access to) that contains the configuration objects which are needed to configure the source business process. It includes details about tables, fields which require values, regions for which the configuration is applicable and definition of a template hierarchy. While business process modeler 138 and business system

configuration tool 142 are two separate tools, they are linked together, or integrated, in a way that allows the user to choose a service instance of tool 142 to which the representation 174 is to be sent. Tool 142 uses metadata from representation 174 to automatically generate the complete configuration templates, including templates, 5 subtemplates, question groups and questions. The templates can be classified for region, industry and applicable business system version. These steps can be repeated multiple times so that additional components (or business processes) can be configured as easily as the first ones, or updates can be made easily as well.

[0050] In addition, configuration tool 142 can be used to configure other things as well, 10 and not just business processes. For instance, input 174 to tool 142 can include information that describes system settings from business system 100. Tool 142 can be used to configure system settings in the new business system instance using this same template-based approach.

[0051] The configuration tool 142 can be used in different environments as well. For 15 instance, instead of business process modeler 138, a more abstract scenario modeler can be used to model a system at a higher abstraction level than a business process (such as at a factory level). As an example, all roles and their corresponding activities can be modeled, mapped, and synchronized to tool 142. Tool 142 can then be used to generate configuration templates for the modeled system. Those templates can then be used to 20 configure an autonomous system with the same behavior as the modeled system.

[0052] FIG. 4 is a block diagram of architecture 130, shown in FIG. 2, except that its 25 elements are disposed in a cloud computing architecture 500. Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location or configuration of the system that delivers the services. In various embodiments, cloud computing delivers the services over a wide area 30 network, such as the internet, using appropriate protocols. For instance, cloud computing providers deliver applications over a wide area network and they can be accessed through a web browser or any other computing component. Software or components of architecture 130 as well as the corresponding data, can be stored on servers at a remote location. The computing resources in a cloud computing environment can be consolidated at a remote data center location or they can be dispersed. Cloud computing infrastructures can deliver services through shared data centers, even though they appear as a single point of access for the user. Thus, the components and functions described herein can be provided from a service provider at a remote location using a cloud computing

architecture. Alternatively, they can be provided from a conventional server, or they can be installed on client devices directly, or in other ways.

[0053] The description is intended to include both public cloud computing and private cloud computing. Cloud computing (both public and private) provides substantially
5 seamless pooling of resources, as well as a reduced need to manage and configure underlying hardware infrastructure.

[0054] A public cloud is managed by a vendor and typically supports multiple consumers using the same infrastructure. Also, a public cloud, as opposed to a private cloud, can free up the end users from managing the hardware. A private cloud may be
10 managed by the organization itself and the infrastructure is typically not shared with other organizations. The organization still maintains the hardware to some extent, such as installations and repairs, etc.

[0055] In the embodiment shown in FIG. 4, some items are similar to those shown in FIG. 2 and they are similarly numbered. FIG. 4 specifically shows that architecture 130 is
15 located in cloud 502 (which can be public, private, or a combination where portions are public while others are private). Therefore, user 134 uses a user device 132 to access those systems through cloud 502.

[0056] FIG. 4 also depicts another embodiment of a cloud architecture. FIG. 4 shows that it is also contemplated that some elements of architecture 130 are disposed in cloud
20 502 while others are not. By way of example, data store 104 (which can be part of architecture 130) can be disposed outside of cloud 502, and accessed through cloud 502. In another embodiment, business system configuration tool 142 is also outside of cloud 502. Regardless of where they are located, they can be accessed directly by device 132, through a network (either a wide area network or a local area network), they can be hosted
25 at a remote site by a service, or they can be provided as a service through a cloud or accessed by a connection service that resides in the cloud. All of these architectures are contemplated herein.

[0057] In addition, the representation 174 used as input to configuration tool 142 can be obtained from an on-premise system or a cloud-based system. The configuration templates
30 184 can then be used to generate output 158 which can be imported into another cloud-based or on-premise system. Alternatively, the configuration templates 184 can be exported from architecture 130 and imported into another system (cloud-based or on-premise) where they are used for configuration.

[0058] It will also be noted that architecture 130, or portions of it, can be disposed on a wide variety of different devices. Some of those devices include servers, desktop computers, laptop computers, tablet computers, or other mobile devices, such as palm top computers, cell phones, smart phones, multimedia players, personal digital assistants, etc.

5 [0059] FIG. 5 is a simplified block diagram of one illustrative embodiment of a handheld or mobile computing device that can be used as a user's or client's hand held device 16, in which the present system (or parts of it) can be deployed. FIGS. 6-10 are examples of handheld or mobile devices.

[0060] FIG. 5 provides a general block diagram of the components of a client device 16
10 that can run components of architecture 130 or system 100 or that interacts with architecture 130, or both. In the device 16, a communications link 13 is provided that allows the handheld device to communicate with other computing devices and under some embodiments provides a channel for receiving information automatically, such as by scanning. Examples of communications link 13 include an infrared port, a serial/USB port,
15 a cable network port such as an Ethernet port, and a wireless network port allowing communication through one or more communication protocols including General Packet Radio Service (GPRS), LTE, HSPA, HSPA+ and other 3G and 4G radio protocols, 1Xrtt, and Short Message Service, which are wireless services used to provide cellular access to a network, as well as 802.11 and 802.11b (Wi-Fi) protocols, and Bluetooth protocol,
20 which provide local wireless connections to networks.

[0061] Under other embodiments, applications or systems are received on a removable Secure Digital (SD) card that is connected to a SD card interface 15. SD card interface 15 and communication links 13 communicate with a processor 17 (which can also embody processors 146 or 150 from FIG. 2) along a bus 19 that is also connected to memory 21
25 and input/output (I/O) components 23, as well as clock 25 and location system 27.

[0062] I/O components 23, in one embodiment, are provided to facilitate input and output operations. I/O components 23 for various embodiments of the device 16 can include input components such as buttons, touch sensors, multi-touch sensors, optical or video sensors, voice sensors, touch screens, proximity sensors, microphones, tilt sensors,
30 and gravity switches and output components such as a display device, a speaker, and or a printer port. Other I/O components 23 can be used as well.

[0063] Clock 25 illustratively comprises a real time clock component that outputs a time and date. It can also, illustratively, provide timing functions for processor 17.

[0064] Location system 27 illustratively includes a component that outputs a current geographical location of device 16. This can include, for instance, a global positioning system (GPS) receiver, a LORAN system, a dead reckoning system, a cellular triangulation system, or other positioning system. It can also include, for example, mapping software or navigation software that generates desired maps, navigation routes and other geographic functions.

[0065] Memory 21 stores operating system 29, network settings 31, applications 33, application configuration settings 35, data store 37, communication drivers 39, and communication configuration settings 41. Memory 21 can include all types of tangible volatile and non-volatile computer-readable memory devices. It can also include computer storage media (described below). Memory 21 stores computer readable instructions that, when executed by processor 17, cause the processor to perform computer-implemented steps or functions according to the instructions. Similarly, device 16 can have a client business system 24 which can run various business applications or embody parts or all of business system 100 or architecture 130. Processor 17 can be activated by other components to facilitate their functionality as well.

[0066] Examples of the network settings 31 include things such as proxy information, Internet connection information, and mappings. Application configuration settings 35 include settings that tailor the application for a specific enterprise or user. Communication configuration settings 41 provide parameters for communicating with other computers and include items such as GPRS parameters, SMS parameters, connection user names and passwords.

[0067] Applications 33 can be applications that have previously been stored on the device 16 or applications that are installed during use, although these can be part of operating system 29, or hosted external to device 16, as well.

[0068] FIG. 6 shows one embodiment in which device 16 is a tablet computer 600. In FIG. 6, computer 600 is shown with user interface display from FIG. 3F displayed on the display screen 602. Screen 602 can be a touch screen (so touch gestures from a user's finger 604 can be used to interact with the application) or a pen-enabled interface that receives inputs from a pen or stylus. It can also use an on-screen virtual keyboard. Of course, it might also be attached to a keyboard or other user input device through a suitable attachment mechanism, such as a wireless link or USB port, for instance. Computer 600 can also illustratively receive voice inputs as well.

[0069] FIGS. 7 and 8 provide additional examples of devices 16 that can be used, although others can be used as well. In FIG. 7, a feature phone, smart phone or mobile phone 45 is provided as the device 16. Phone 45 includes a set of keypads 47 for dialing phone numbers, a display 49 capable of displaying images including application images, icons, web pages, photographs, and video, and control buttons 51 for selecting items shown on the display. The phone includes an antenna 53 for receiving cellular phone signals such as General Packet Radio Service (GPRS) and 1Xrtt, and Short Message Service (SMS) signals. In some embodiments, phone 45 also includes a Secure Digital (SD) card slot 55 that accepts a SD card 57.

10 [0070] The mobile device of FIG. 8 is a personal digital assistant (PDA) 59 or a multimedia player or a tablet computing device, etc. (hereinafter referred to as PDA 59). PDA 59 includes an inductive screen 61 that senses the position of a stylus 63 (or other pointers, such as a user's finger) when the stylus is positioned over the screen. This allows the user to select, highlight, and move items on the screen as well as draw and write. PDA
15 59 also includes a number of user input keys or buttons (such as button 65) which allow the user to scroll through menu options or other display options which are displayed on display 61, and allow the user to change applications or select user input functions, without contacting display 61. Although not shown, PDA 59 can include an internal antenna and an infrared transmitter/receiver that allow for wireless communication with
20 other computers as well as connection ports that allow for hardware connections to other computing devices. Such hardware connections are typically made through a cradle that connects to the other computer through a serial or USB port. As such, these connections are non-network connections. In one embodiment, mobile device 59 also includes a SD card slot 67 that accepts a SD card 69.

25 [0071] FIG. 9 is similar to FIG. 7 except that the phone is a smart phone 71. Smart phone 71 has a touch sensitive display 73 that displays icons or tiles or other user input mechanisms 75. Mechanisms 75 can be used by a user to run applications, make calls, perform data transfer operations, etc. In general, smart phone 71 is built on a mobile operating system and offers more advanced computing capability and connectivity than a
30 feature phone. FIG. 10 shows smart phone 71 with the display of FIG. 3G displayed on it.

[0072] Note that other forms of the devices 16 are possible.

[0073] FIG. 11 is one embodiment of a computing environment in which architecture 130, or system 100, or parts of it, (for example) can be deployed. With reference to FIG. 11, an exemplary system for implementing some embodiments includes a general-purpose

computing device in the form of a computer 810. Components of computer 810 may include, but are not limited to, a processing unit 820 (which can comprise processor 102, 146 or 150), a system memory 830, and a system bus 821 that couples various system components including the system memory to the processing unit 820. The system bus 821
5 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral
10 Component Interconnect (PCI) bus also known as Mezzanine bus. Memory and programs described with respect to FIG. 2 can be deployed in corresponding portions of FIG. 11.

[0074] Computer 810 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 810 and includes both volatile and nonvolatile media, removable and non-removable
15 media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media is different from, and does not include, a modulated data signal or carrier wave. It includes hardware storage media including both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as
20 computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information
25 and which can be accessed by computer 810. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not
30 limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0075] The system memory 830 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 831 and random access memory (RAM) 832. A basic input/output system 833 (BIOS), containing the basic routines that help to transfer information between elements within computer 810, such as
5 during start-up, is typically stored in ROM 831. RAM 832 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 820. By way of example, and not limitation, FIG. 11 illustrates operating system 834, application programs 835, other program modules 836, and program data 837.

[0076] The computer 810 may also include other removable/non-removable
10 volatile/nonvolatile computer storage media. By way of example only, FIG. 11 illustrates a hard disk drive 841 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 851 that reads from or writes to a removable, nonvolatile magnetic disk 852, and an optical disk drive 855 that reads from or writes to a removable, nonvolatile optical disk 856 such as a CD ROM or other optical media. Other
15 removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 841 is typically connected to the system bus 821 through a non-removable memory interface such as interface 840, and magnetic
20 disk drive 851 and optical disk drive 855 are typically connected to the system bus 821 by a removable memory interface, such as interface 850.

[0077] Alternatively, or in addition, the functionality described herein can be performed, at least in part, by one or more hardware logic components. For example, and without limitation, illustrative types of hardware logic components that can be used include Field-
25 programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), etc.

[0078] The drives and their associated computer storage media discussed above and illustrated in FIG. 11, provide storage of computer readable instructions, data structures,
30 program modules and other data for the computer 810. In FIG. 11, for example, hard disk drive 841 is illustrated as storing operating system 844, application programs 845, other program modules 846, and program data 847. Note that these components can either be the same as or different from operating system 834, application programs 835, other program modules 836, and program data 837. Operating system 844, application programs

845, other program modules 846, and program data 847 are given different numbers here to illustrate that, at a minimum, they are different copies.

[0079] A user may enter commands and information into the computer 810 through input devices such as a keyboard 862, a microphone 863, and a pointing device 861, such as a mouse, trackball or touch pad. Other input devices (not shown) may include a joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 820 through a user input interface 860 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A visual display 891 or other type of display device is also connected to the system bus 821 via an interface, such as a video interface 890. In addition to the monitor, computers may also include other peripheral output devices such as speakers 897 and printer 896, which may be connected through an output peripheral interface 895.

[0080] The computer 810 is operated in a networked environment using logical connections to one or more remote computers, such as a remote computer 880. The remote computer 880 may be a personal computer, a hand-held device, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 810. The logical connections depicted in FIG. 11 include a local area network (LAN) 871 and a wide area network (WAN) 873, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0081] When used in a LAN networking environment, the computer 810 is connected to the LAN 871 through a network interface or adapter 870. When used in a WAN networking environment, the computer 810 typically includes a modem 872 or other means for establishing communications over the WAN 873, such as the Internet. The modem 872, which may be internal or external, may be connected to the system bus 821 via the user input interface 860, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 810, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 11 illustrates remote application programs 885 as residing on remote computer 880. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0082] It should also be noted that the different embodiments described herein can be combined in different ways. That is, parts of one or more embodiments can be combined with parts of one or more other embodiments. All of this is contemplated herein.

[0083] Although the subject matter has been described in language specific to structural
5 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

1. A computer-implemented method, comprising:
obtaining a representation of a source process having process functionality;
translating the representation of the source process to a translated representation;
identifying a configuration tool instance that is to be used; and
generating a configuration template, with the identified configuration tool instance,
from the translated representation, with user configuration input
mechanisms that receive user configuration inputs to configure a system
instance with the process functionality.
2. The computer-implemented method of claim 1 wherein the source process
comprises a business process and the system instance comprises a business system
instance, and wherein obtaining a representation of the source business process comprises:
obtaining a first representation of the source business process at a business process
modeler.
3. The computer-implemented method of claim 2 wherein identifying a configuration
tool instance, comprises:
receiving a user synchronization input at the business process modeler to send the
representation of the source business process to the identified configuration
tool instance; and
displaying a selection user input mechanism at the business process modeler that
receives user selection of the identified configuration tool instance.
4. The computer-implemented method of claim 2 displaying a selection user input
mechanism comprises:
displaying a list of user actuable input mechanisms, each corresponding to a
different configuration tool instance; and
receiving user actuation of a given one of the user actuable input mechanisms.

5. The computer-implemented method of claim 4 wherein generating a configuration template comprises:
 - receiving the translated representation at the configuration tool instance
 - corresponding to given user actuatable input mechanism actuated by the user.

6. The computer-implemented method of claim 5 wherein the configuration tool instance comprises an instance of a question-based business system configuration tool, and wherein generating a configuration template comprises:
 - generating a set of questions, in question groups that each comprise a set of questions that are related by subject matter, that are answered by the user configuration inputs.

7. The computer-implemented method of claim 6 wherein translating comprises:
 - translating the recorded representation of the source business process into a representation having a schema expected by the configuration tool instance.

8. The computer-implemented method of claim 2 and further comprising:
 - receiving the user configuration inputs to obtain a configured configuration template; and
 - applying the configured configuration template to the business system instance to obtain a configured instance of the business system that has the process functionality of the source business process.

9. A configuration system, comprising:
 - an input component that provides a representation of a process that has process functionality;
 - a system configuration tool that receives the representation of the process and generates a configuration template with user input mechanisms that receive user configuration inputs and that generates a configured system instance that has the process functionality based on the user configuration inputs to the configuration template; and
 - a computer processor that is a functional part of the system and is activated by the input component and the system configuration tool to facilitate providing

the representation, generating the configuration template and generating the configured system instance.

10. A computer readable storage medium that stores computer readable instructions which, when executed by a computer, cause the computer to perform steps, comprising:
 - obtaining a first representation of a source process having process functionality;
 - translating the first representation of the source process to a second representation;
 - identifying a configuration tool instance that is to be used;
 - sending the second representation of the source business process to the identified configuration tool instance; and
 - generating a configuration template, with the identified configuration tool instance, from the second representation, with user configuration input mechanisms that receive user configuration inputs to configure a system instance with the process functionality.

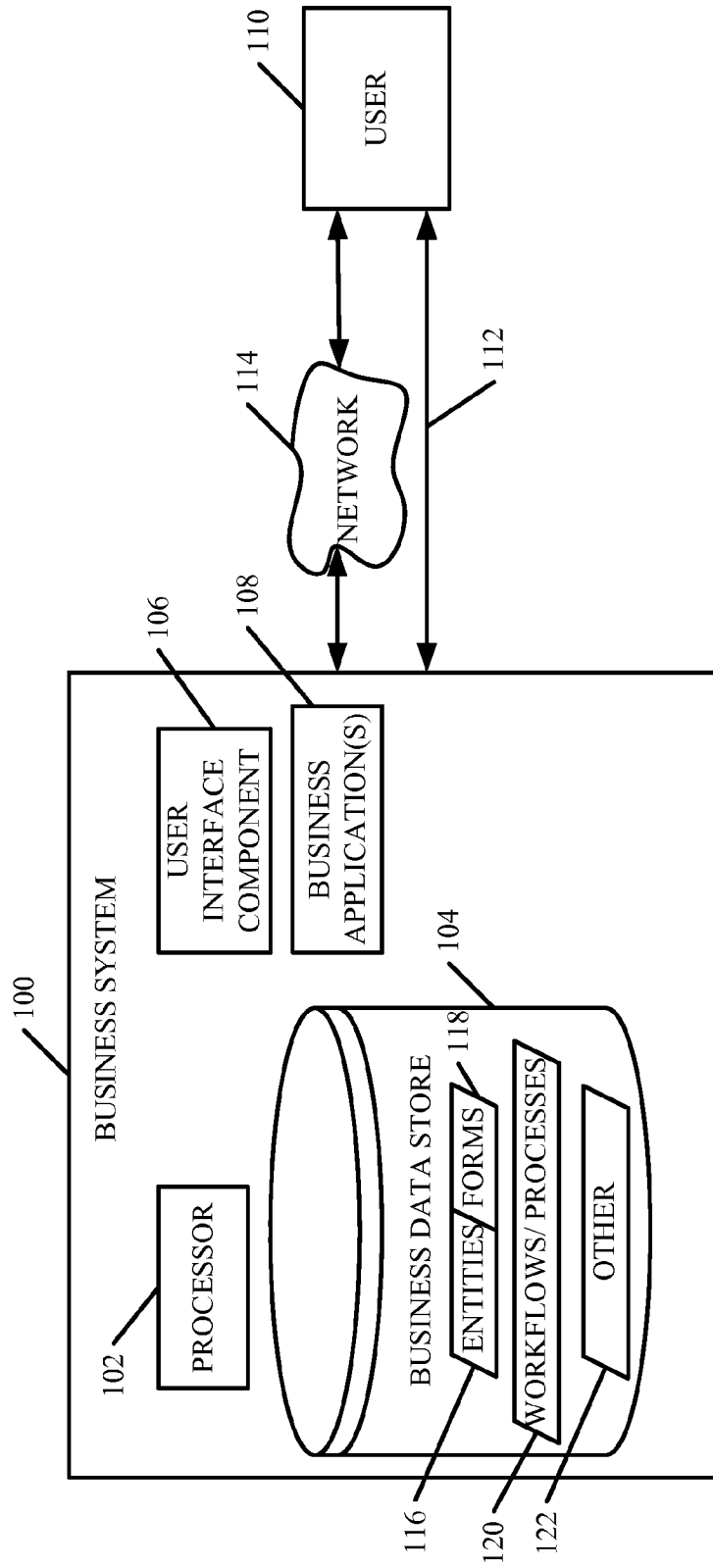


FIG. 1

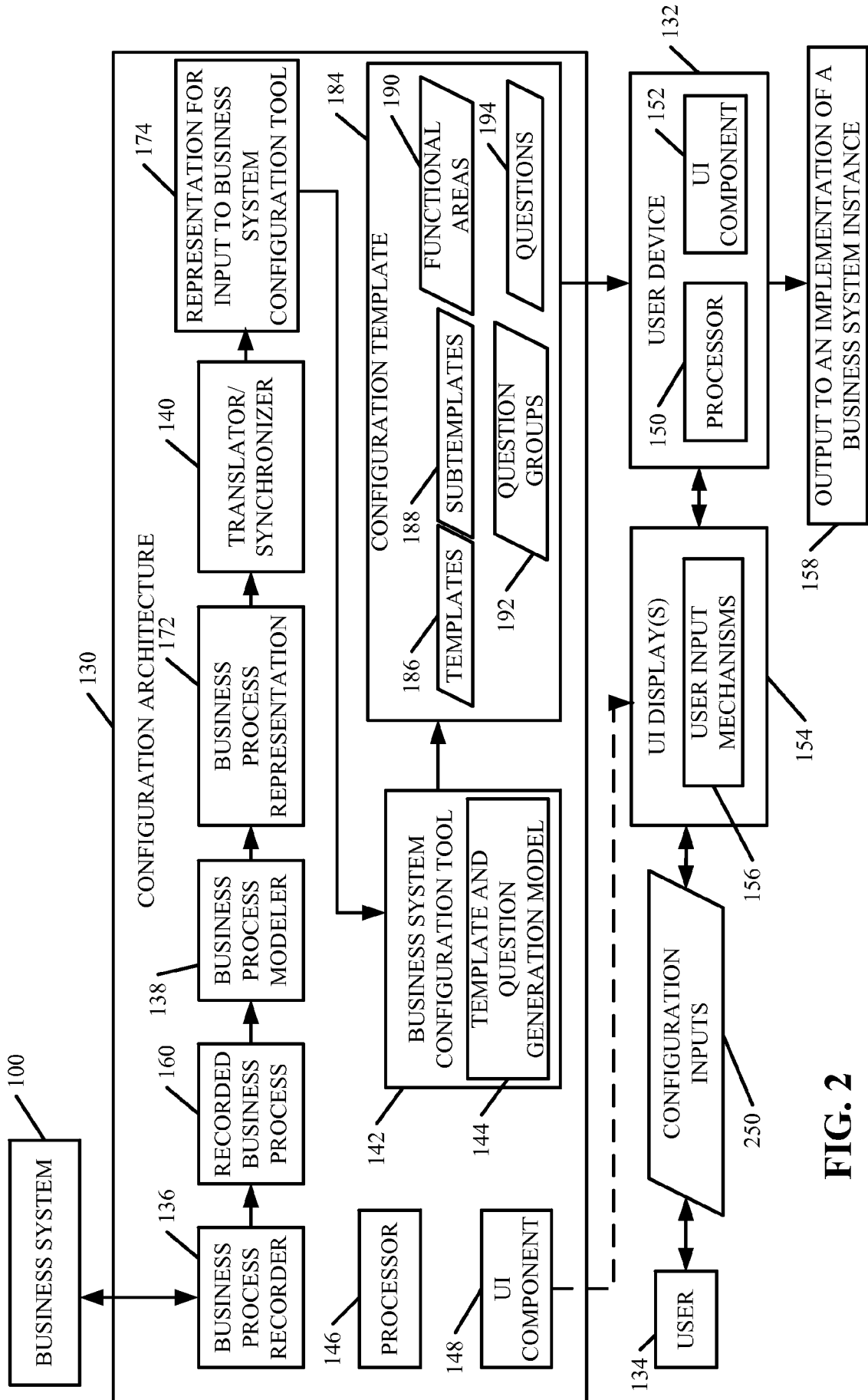


FIG. 2

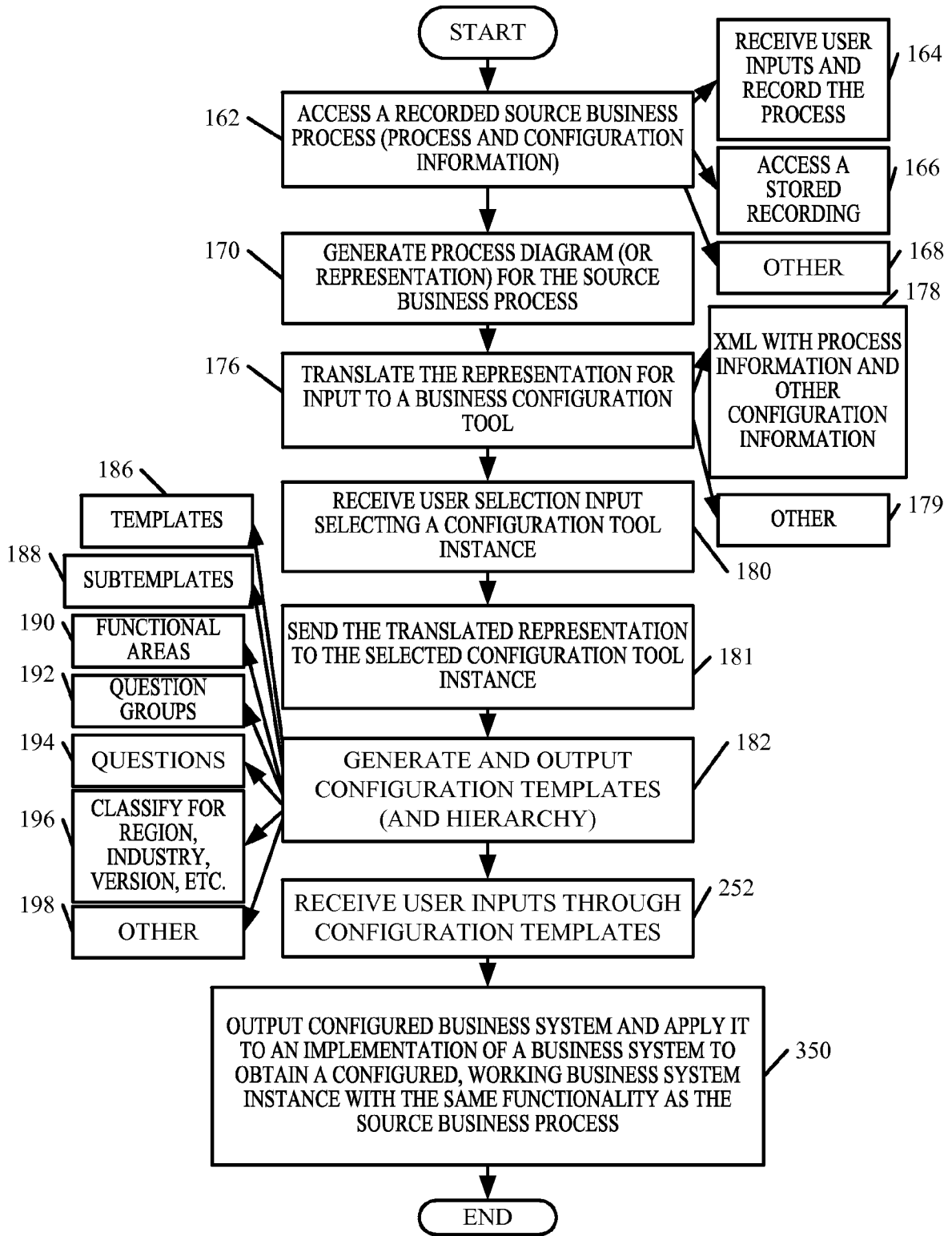


FIG. 3

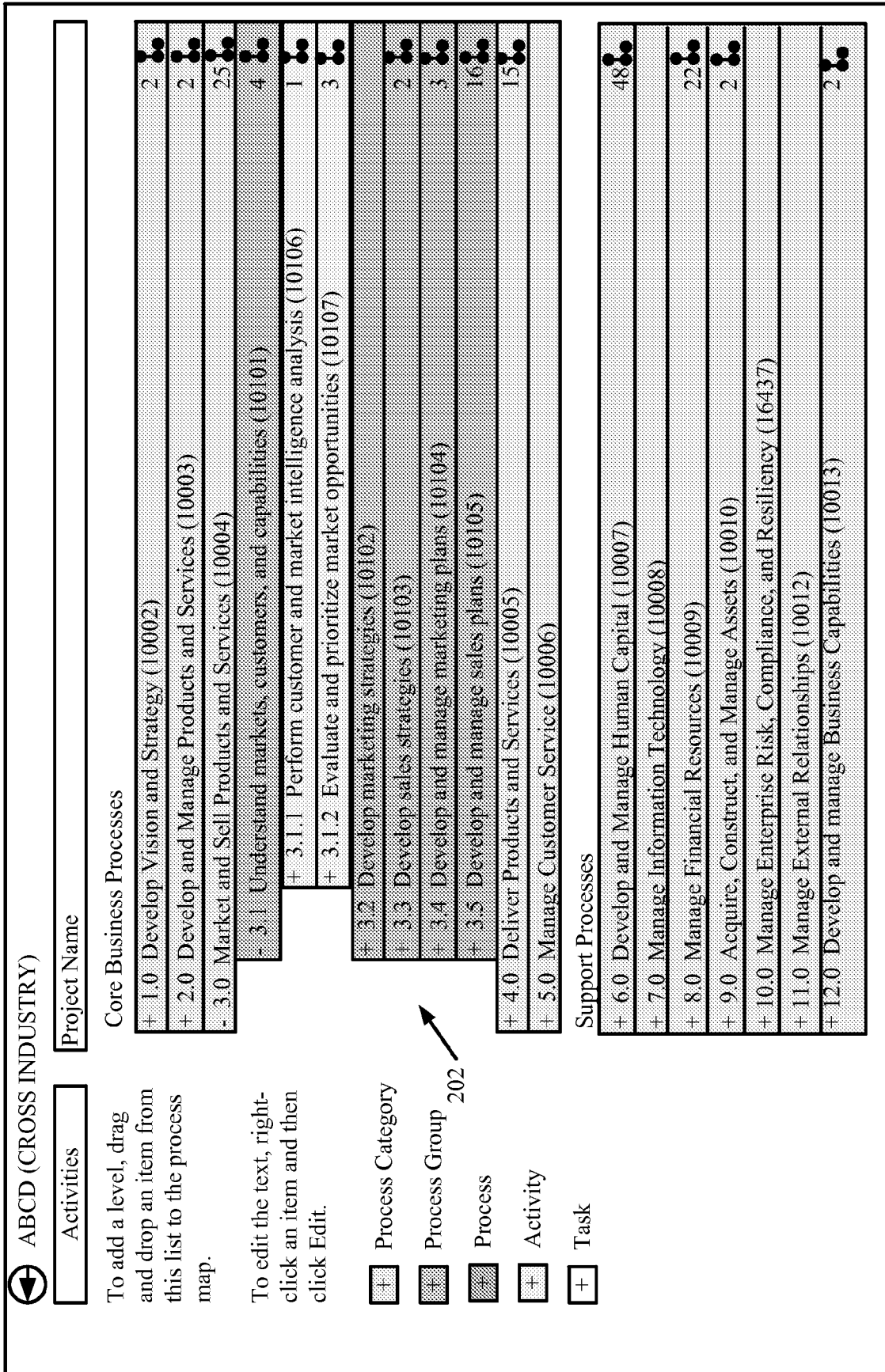


FIG. 3A

ProjectName					
<h3>Synchronize with Configuration Tool</h3>					
Created at	Created by	Solution name	End point	Status	Status detail
6/14/2013 10:58 AM	Doe, John	ABCD	USSC	Completed	Success: 118
<>> Page 1 of 1 >>					
<p>Select a configuration system from the list and click "Call"</p>					
Specific Configuration System			Call		

204

206

FIG. 3B

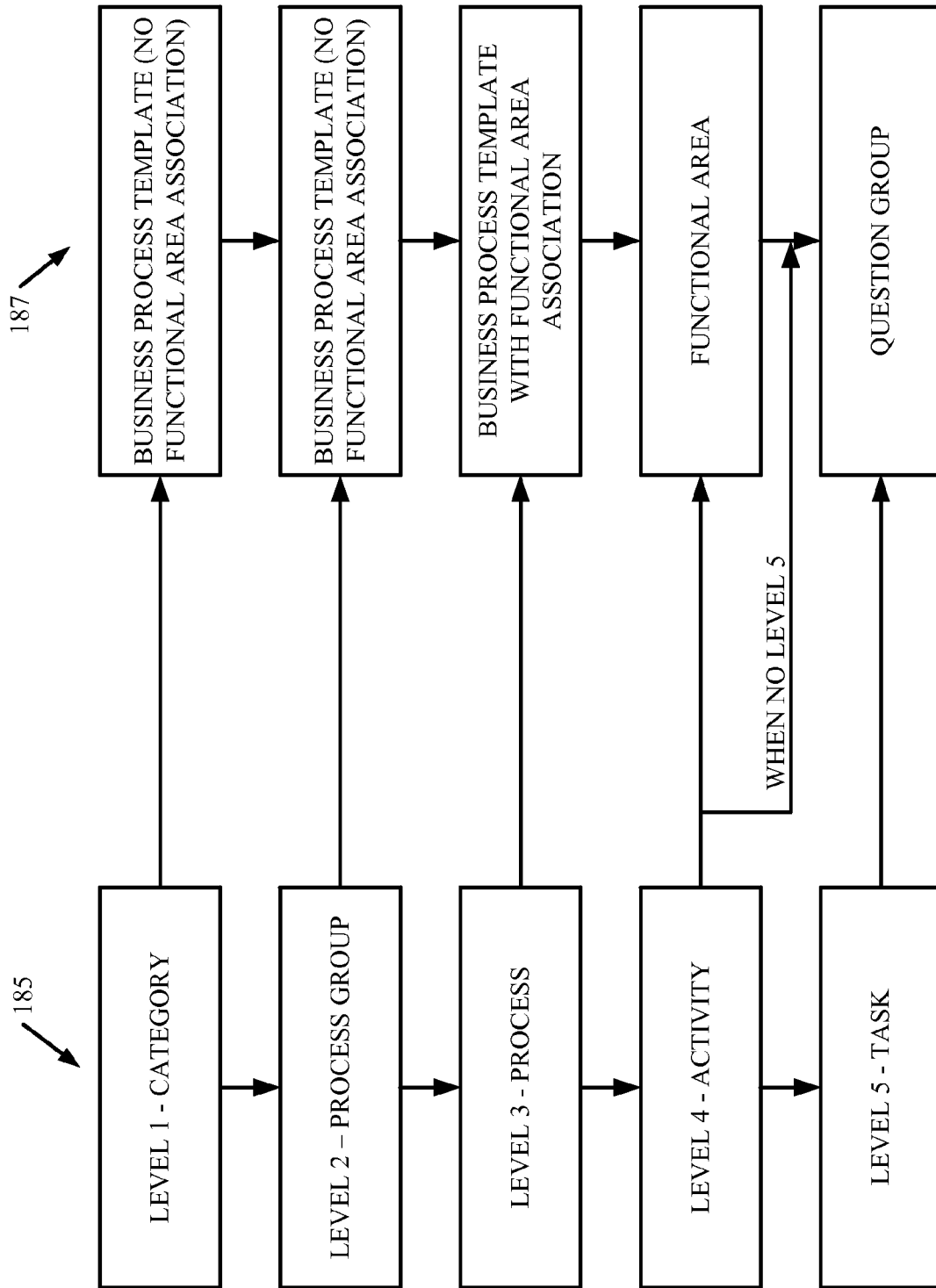


FIG. 3C

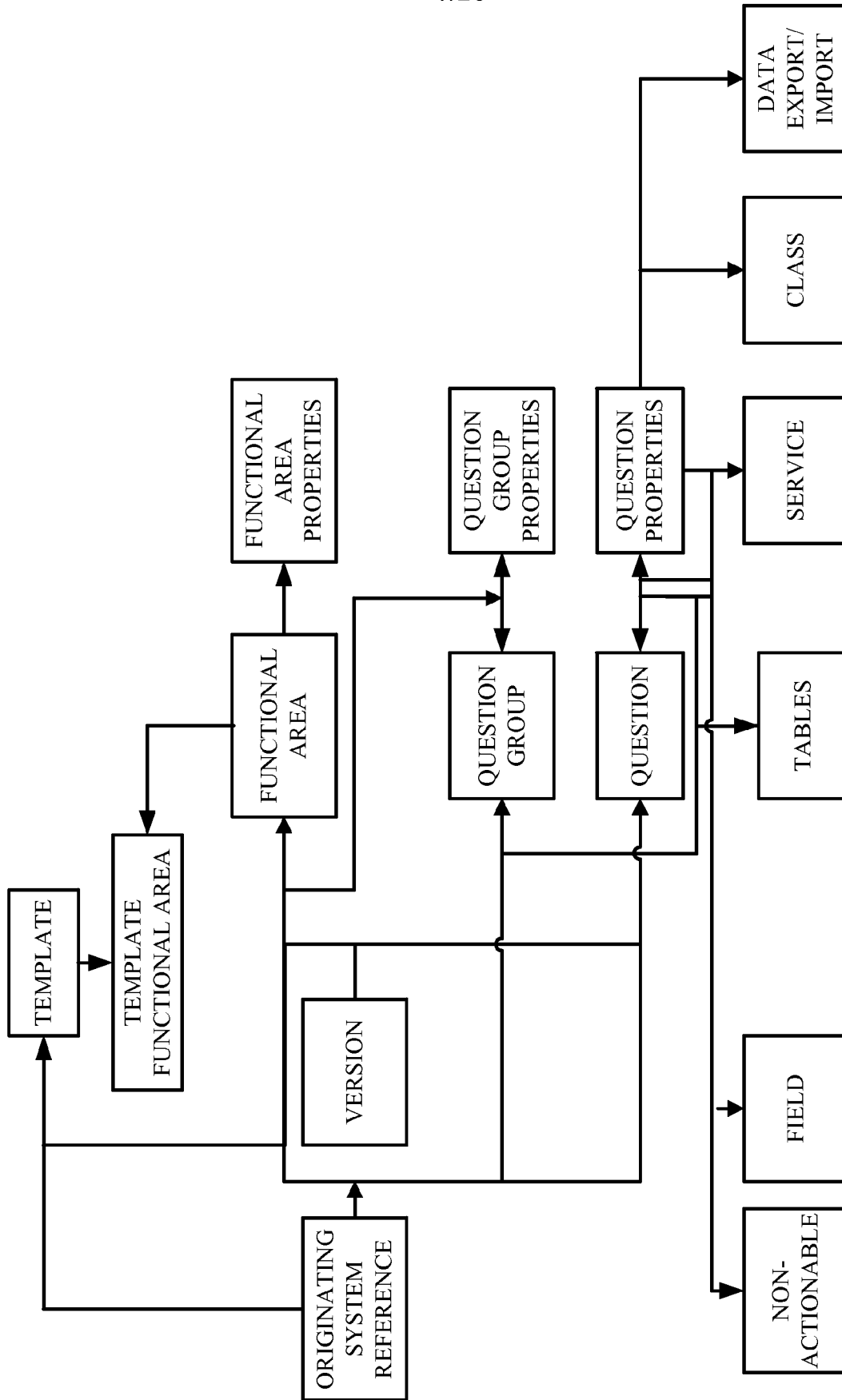


FIG. 3D

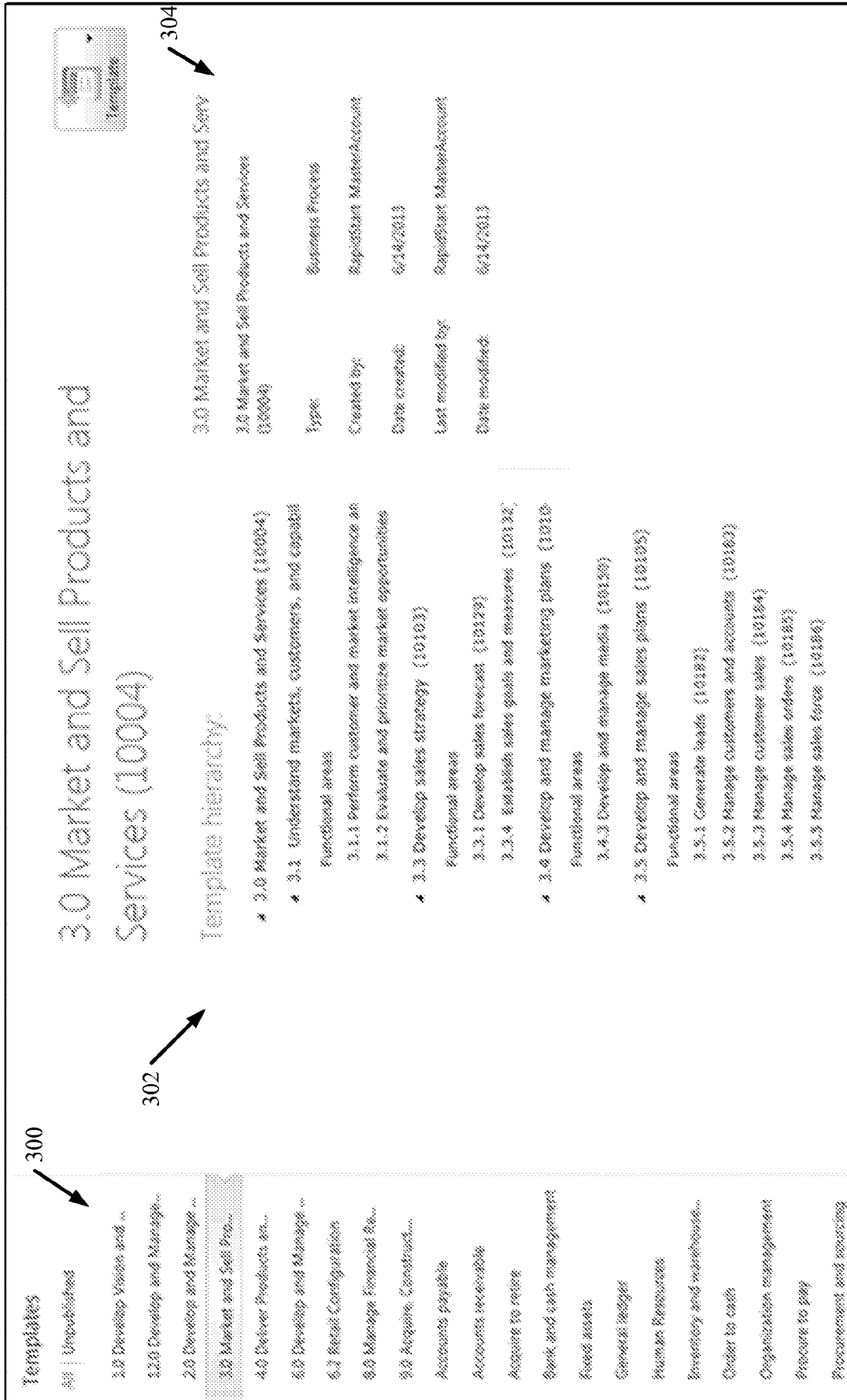


FIG. 3E

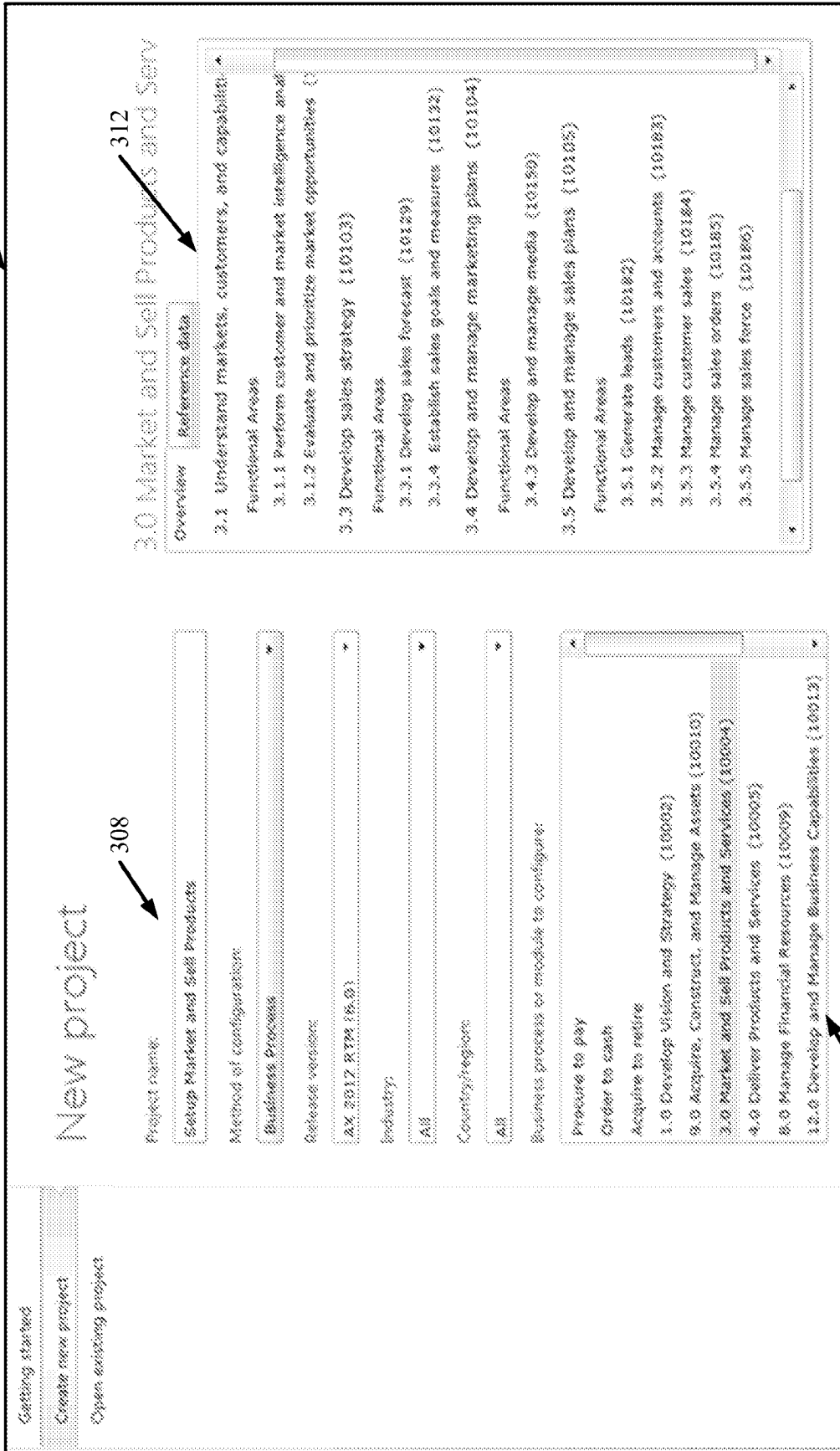


FIG. 3F

310

Basic Information

Question name:

Question text:

Question description:

Is this a required question?
 No Yes

Is this question displayed?
 No Yes

Table name:

Field(s):
 AccountID
 BankGroupId
 CurrencyCode
 Name

Generate workbook

Assign template: Upload file Download assigned workbook

FIG. 3G

Question name:	DepreciationMin
Question text:	Minimum depreciation amount
Question description:	Minimum depreciation amount when depreciating with reducing balance.
Is this a required question?	<input type="radio"/> No <input type="radio"/> Yes
Is this question displayed?	<input type="radio"/> No <input checked="" type="radio"/> Yes
Table name:	AssetParameter
Field name:	DepreciationMin
Base data type:	Real
Control type:	Textbook

FIG. 3H

Basic Information

Question name:

Question text:

Question description:

Is this a required question? No Yes

Is this question displayed? No Yes

Release version: AX2012 RTM (6.0)

Industry: All Country/region: X

Table name: Add table

Field(s): AbsenceFromDate AbsenceSetupDataAreaID AbsenceSetupId Active

Generate and review template: Generate CSV file

Assign template: Upload CSV Download assigned template

FIG. 3I

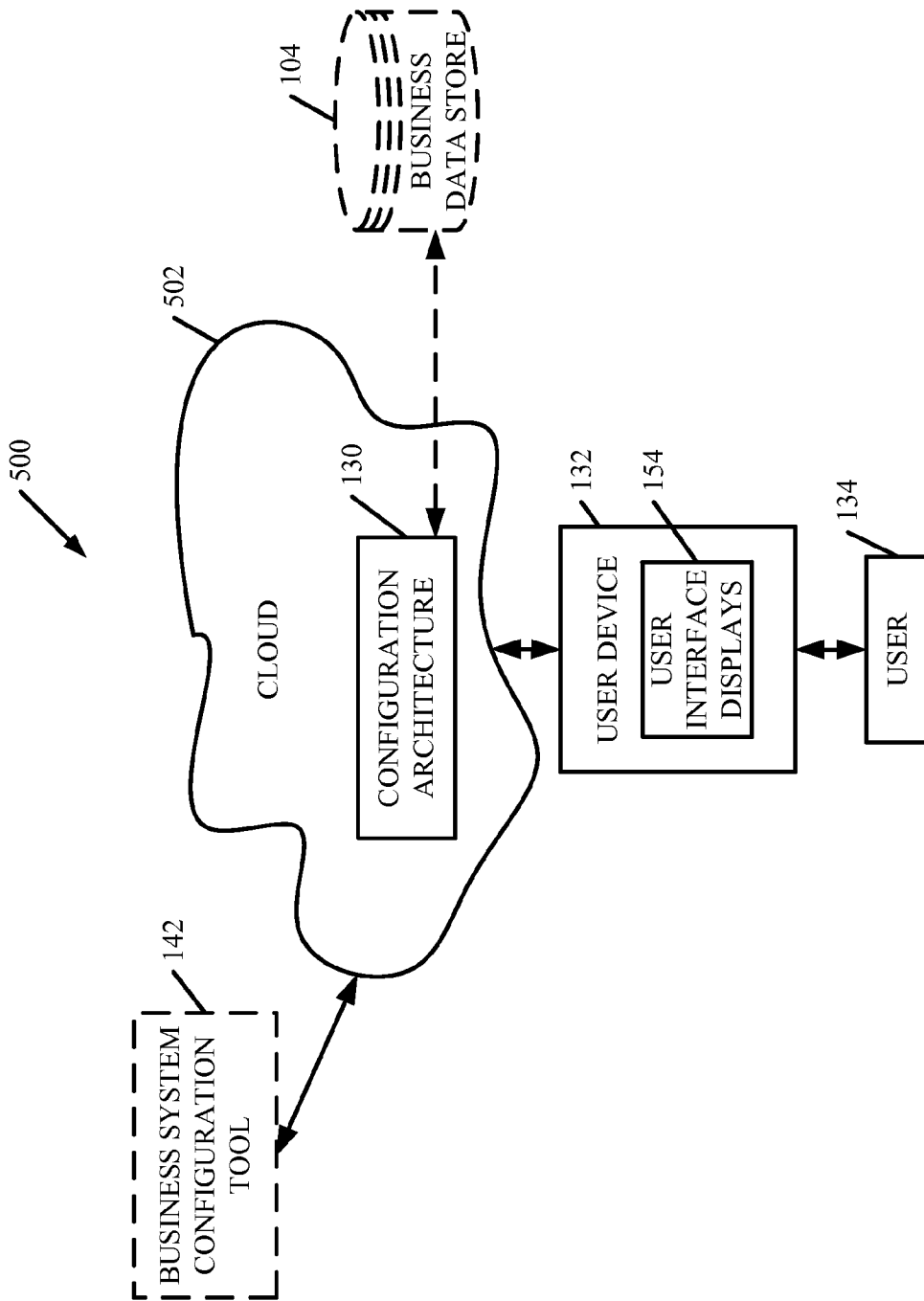


FIG. 4

14/20

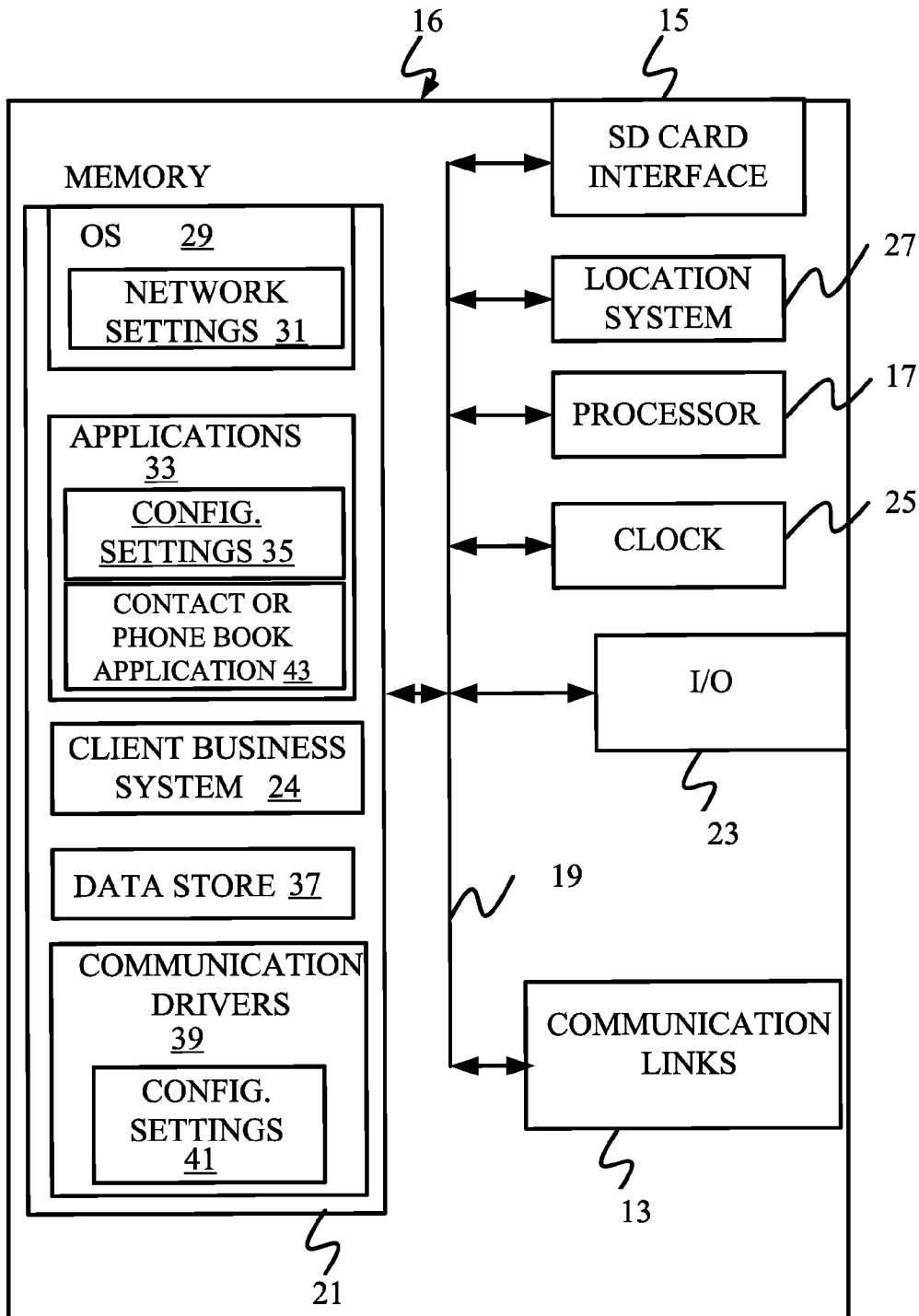


FIG. 5

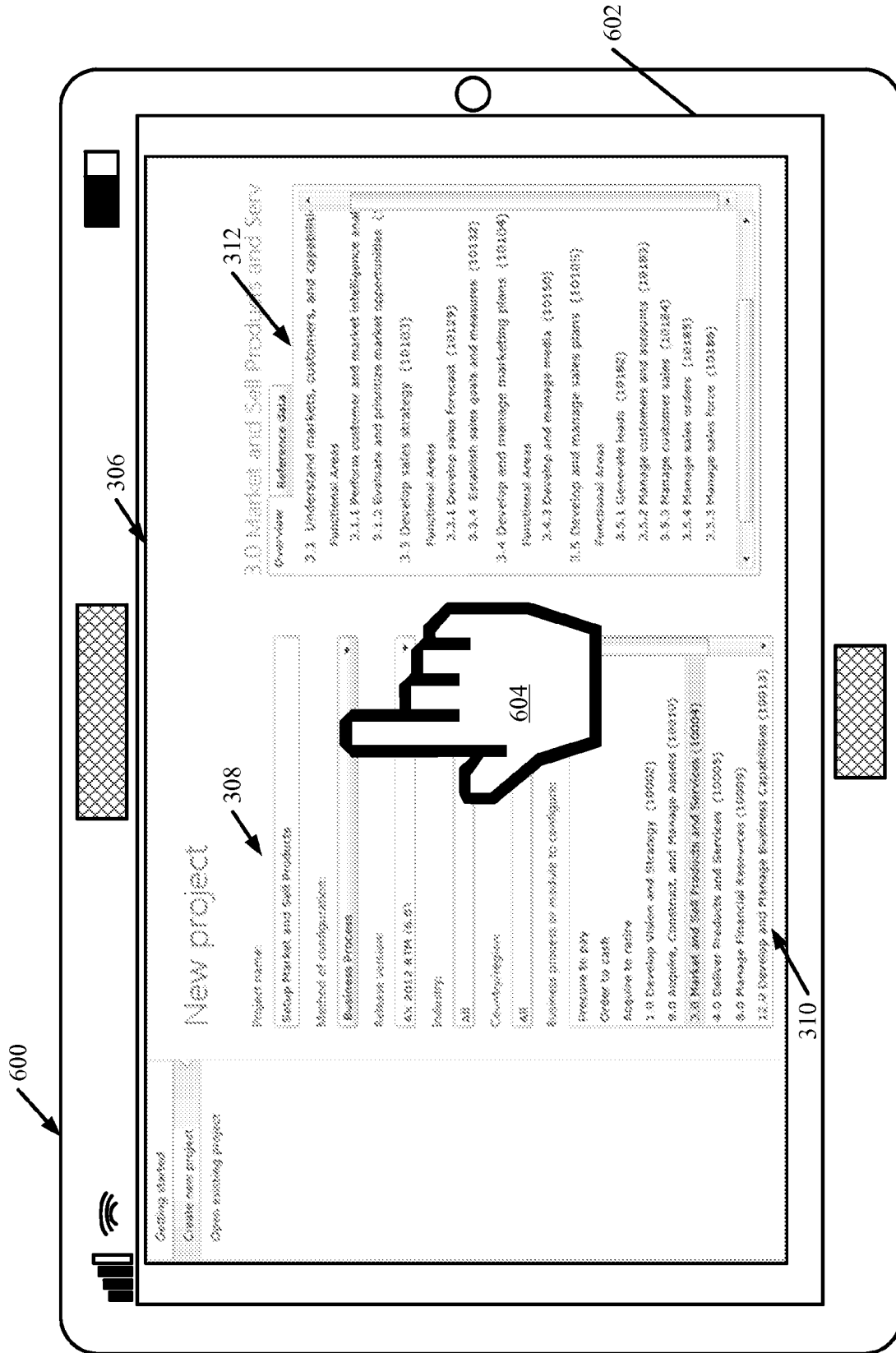


FIG. 6

16/20

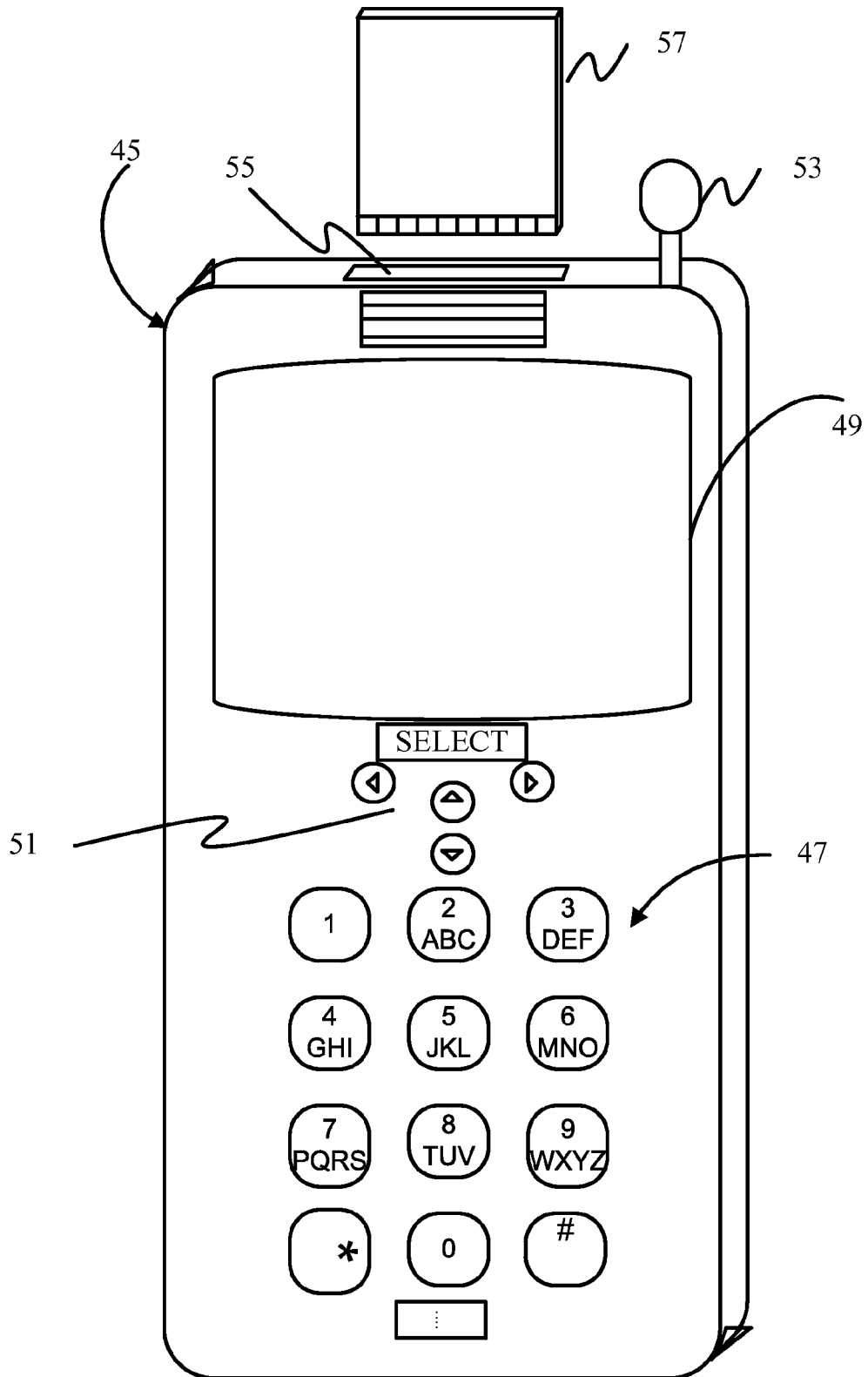


FIG. 7

17/20

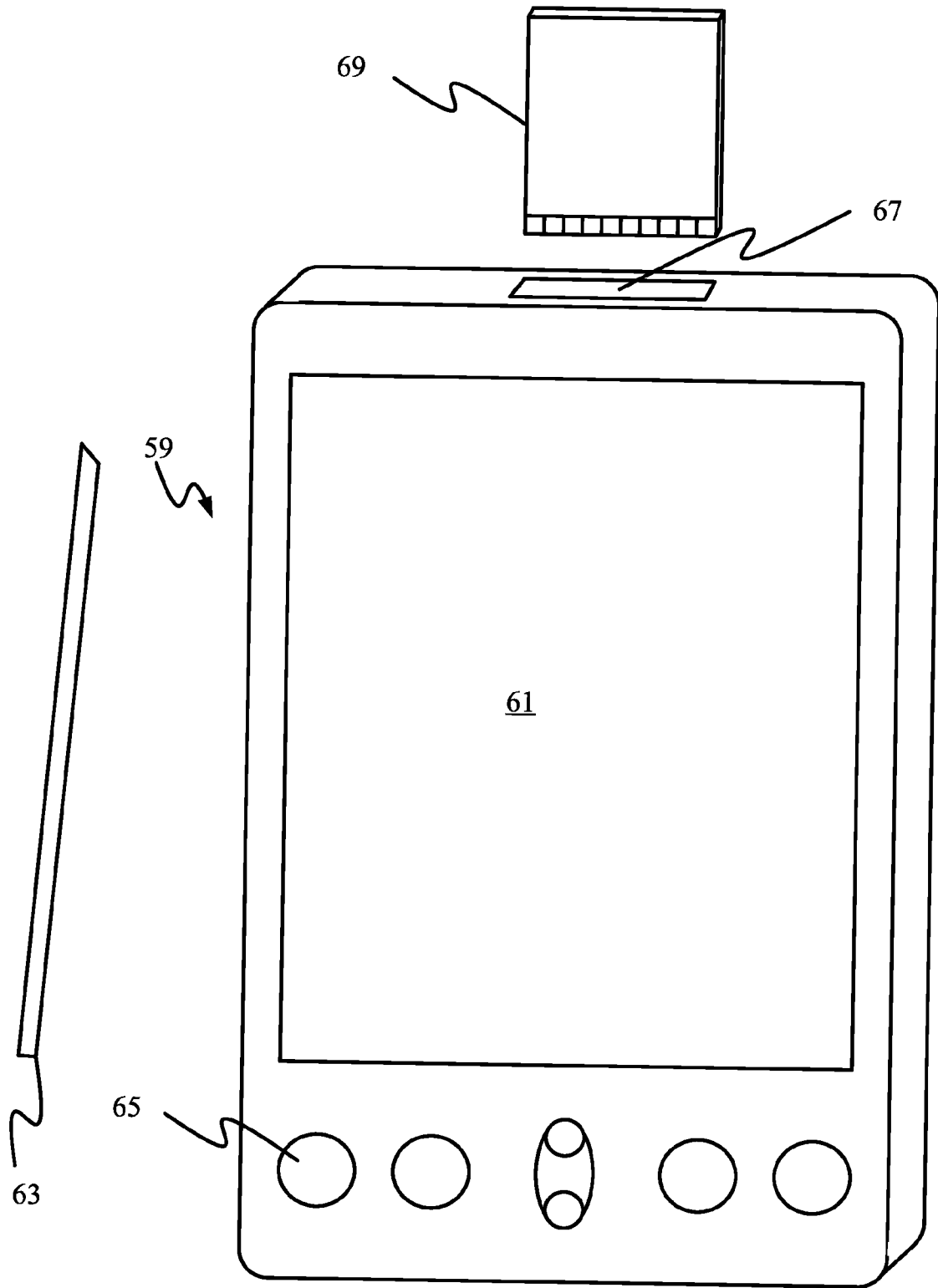


FIG. 8

18/20

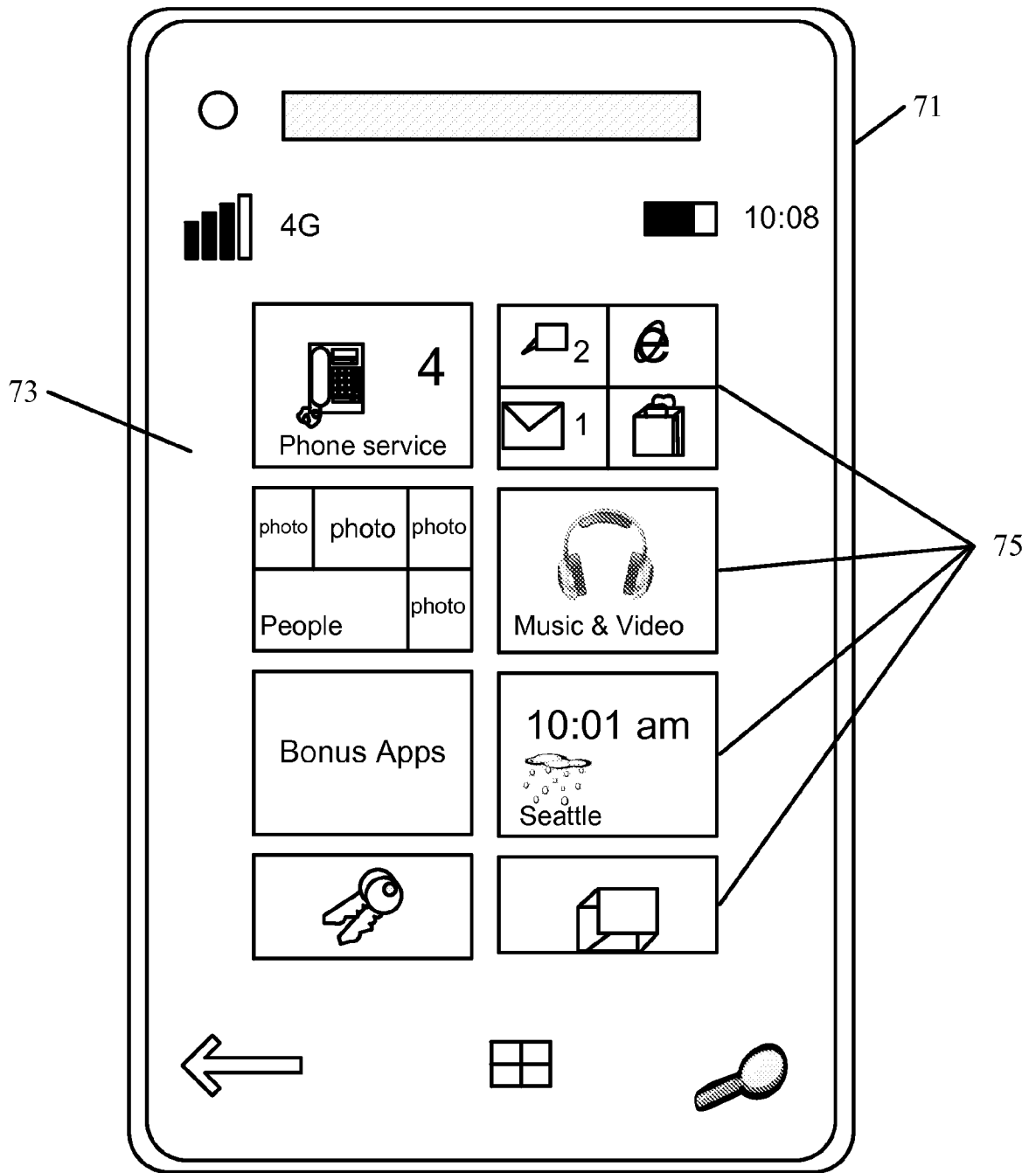


FIG. 9

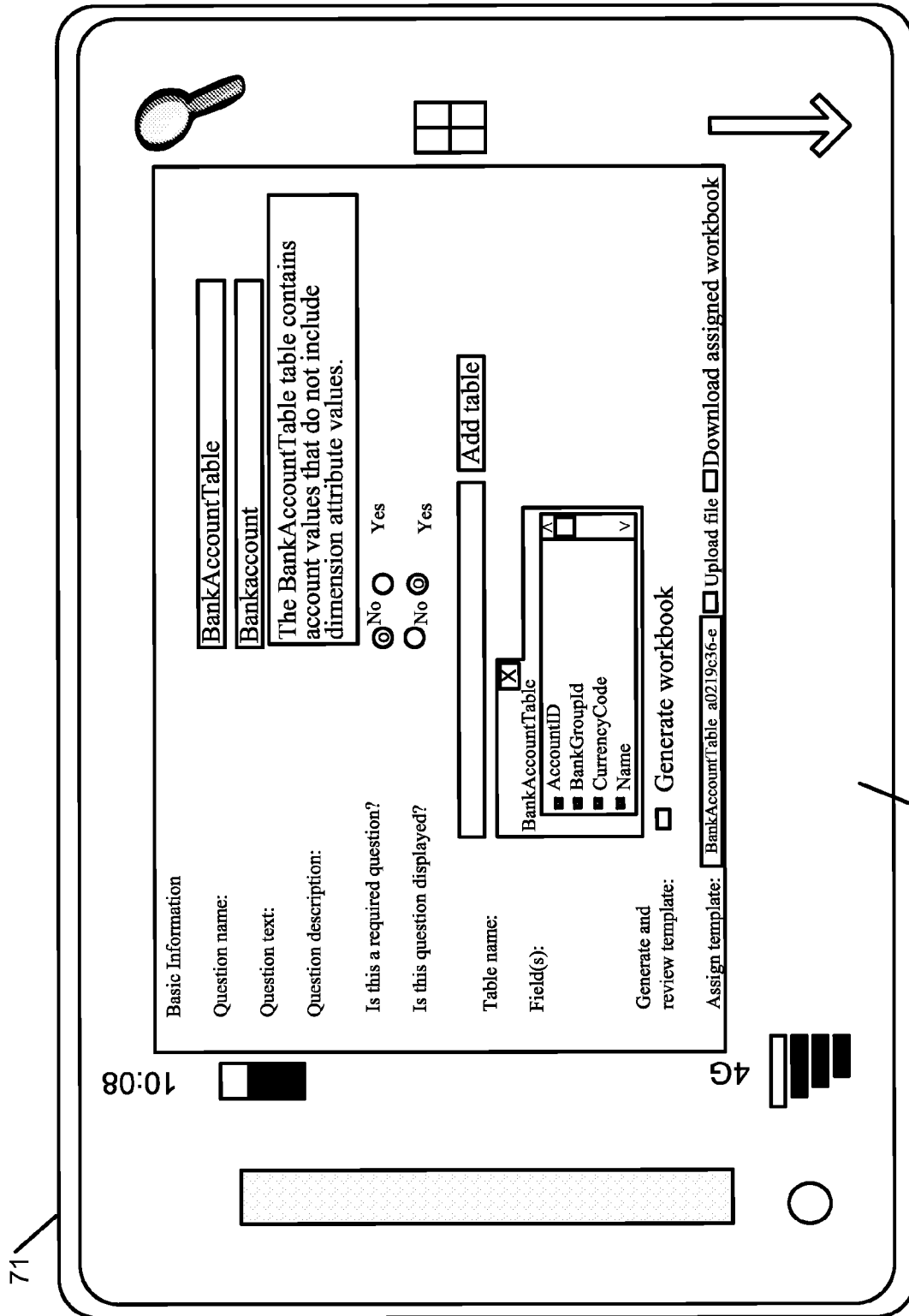


FIG. 10

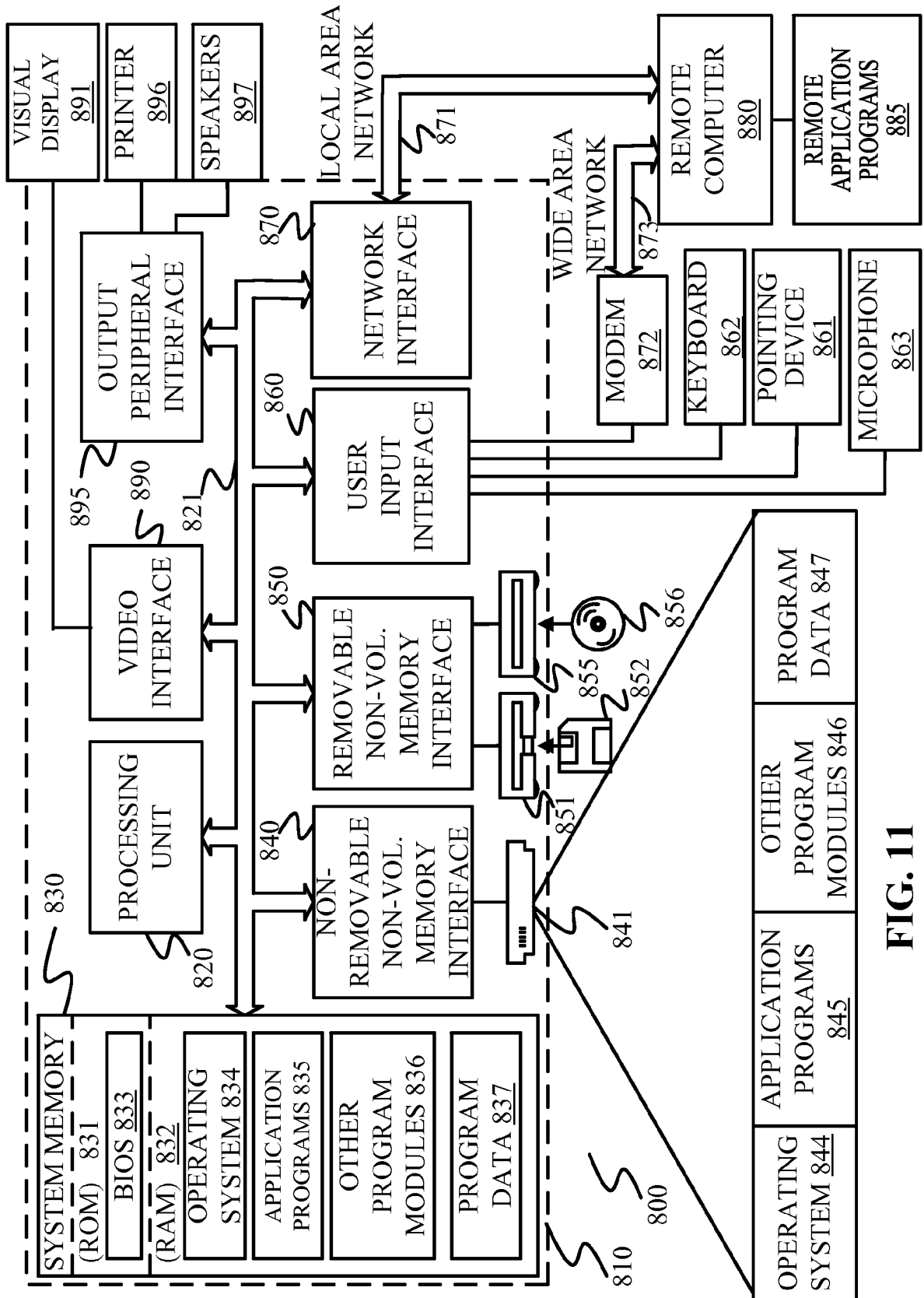


FIG. 11