



US 20150095813A1

(19) **United States**

(12) **Patent Application Publication**
Welicki et al.

(10) **Pub. No.: US 2015/0095813 A1**

(43) **Pub. Date: Apr. 2, 2015**

(54) **MULTI-ACT CREATION USER INTERFACE ELEMENT**

(71) Applicant: **MICROSOFT CORPORATION**,
Redmond, WA (US)

(72) Inventors: **Leon Ezequiel Welicki**, Issaquah, WA (US); **Brad Olenick**, Redmond, WA (US); **Vishal R. Joshi**, Redmond, WA (US); **Andrew Birck**, Seattle, WA (US); **Alvaro Rahul Dias**, Bellevue, WA (US); **Guru Kumaraguru**, Redmond, WA (US); **Federico Silva Armas**, Seattle, WA (US); **Stephen Michael Danton**, Seattle, WA (US); **Brendyn Alexander**, Seattle, WA (US); **David Anson**, Snoqualmie, WA (US); **Jean-Sebastien Goupil**, Redmond, WA (US); **Bradley D. Millington**, Bellevue, WA (US); **Adam Abdelhamed**, Bellevue, WA (US); **Jesse David Francisco**, Lake Stevens, WA (US); **Wai Man Yuen**, Bellevue, WA (US); **Jon Harris**, Sammamish, WA (US); **Jonah Bush Sterling**, Seattle, WA (US); **Karandeep Singh Anand**, Redmond, WA (US); **William J. Staples**, Duvall, WA (US); **Madhur Joshi**, Kirkland, WA (US); **Nathan J. Totten**, Bellevue, WA (US)

(21) Appl. No.: **14/231,883**

(22) Filed: **Apr. 1, 2014**

Related U.S. Application Data

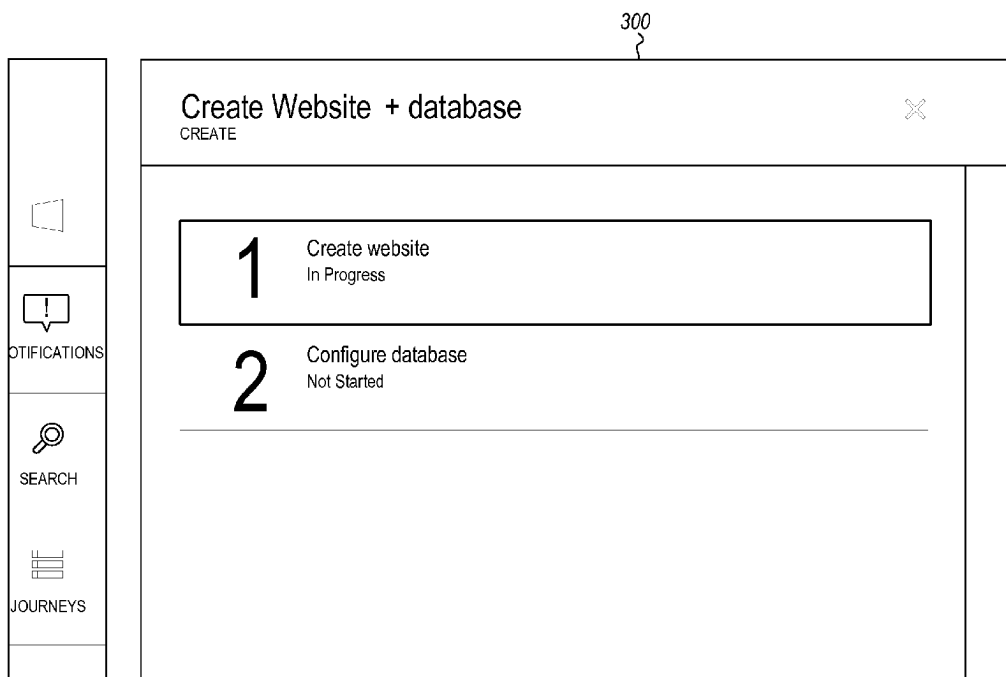
(60) Provisional application No. 61/905,128, filed on Nov. 15, 2013, provisional application No. 61/884,743, filed on Sep. 30, 2013, provisional application No. 61/905,111, filed on Nov. 15, 2013, provisional application No. 61/905,243, filed on Nov. 17, 2013, provisional application No. 61/905,114, filed on Nov. 15, 2013, provisional application No. 61/905,116, filed on Nov. 15, 2013, provisional application No. 61/905,129, filed on Nov. 15, 2013, provisional application No. 61/905,105, filed on Nov. 15, 2013, provisional application No. 61/905,247, filed on Nov. 17, 2013, provisional application No. 61/905,101, filed on Nov. 15, 2013, provisional application No. 61/905,119, filed on Nov. 15, 2013.

Publication Classification

(51) **Int. Cl.**
G06F 3/0484 (2006.01)
(52) **U.S. Cl.**
CPC **G06F 3/0484** (2013.01)
USPC **715/762**

(57) **ABSTRACT**

A multi-act creation user interface element that is displayed when a user instructs a computing system to initiate creation of a resource that has multiple associated creation processing acts. For each of some or all of the creation processing acts, the multi-act creation user interface element causes a set of one or more controls to appear, which the user may then enter creation information into. The multi-step creation user interface element persists creation information so entered so that the creation information is available for further creation processing acts.



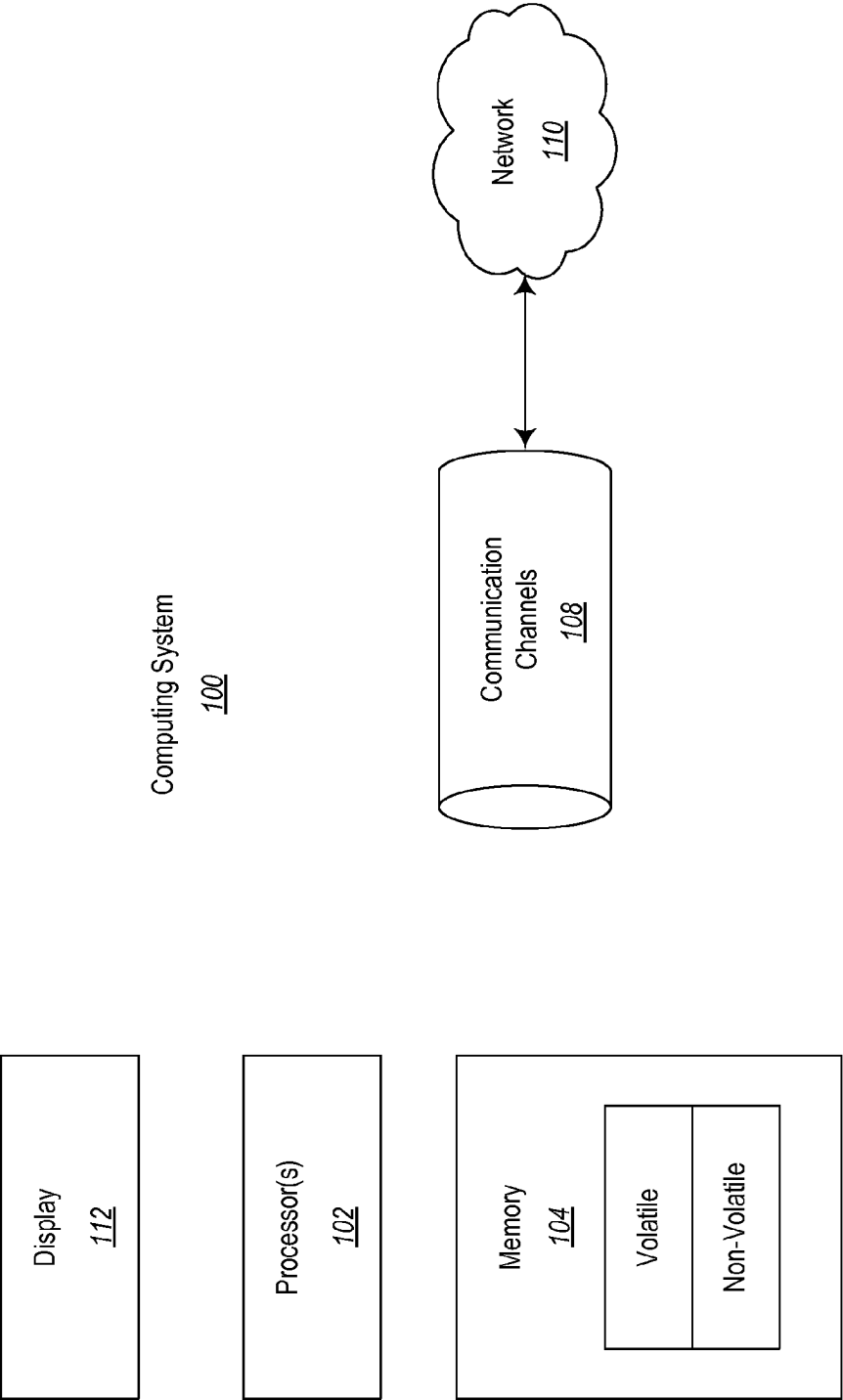


Figure 1

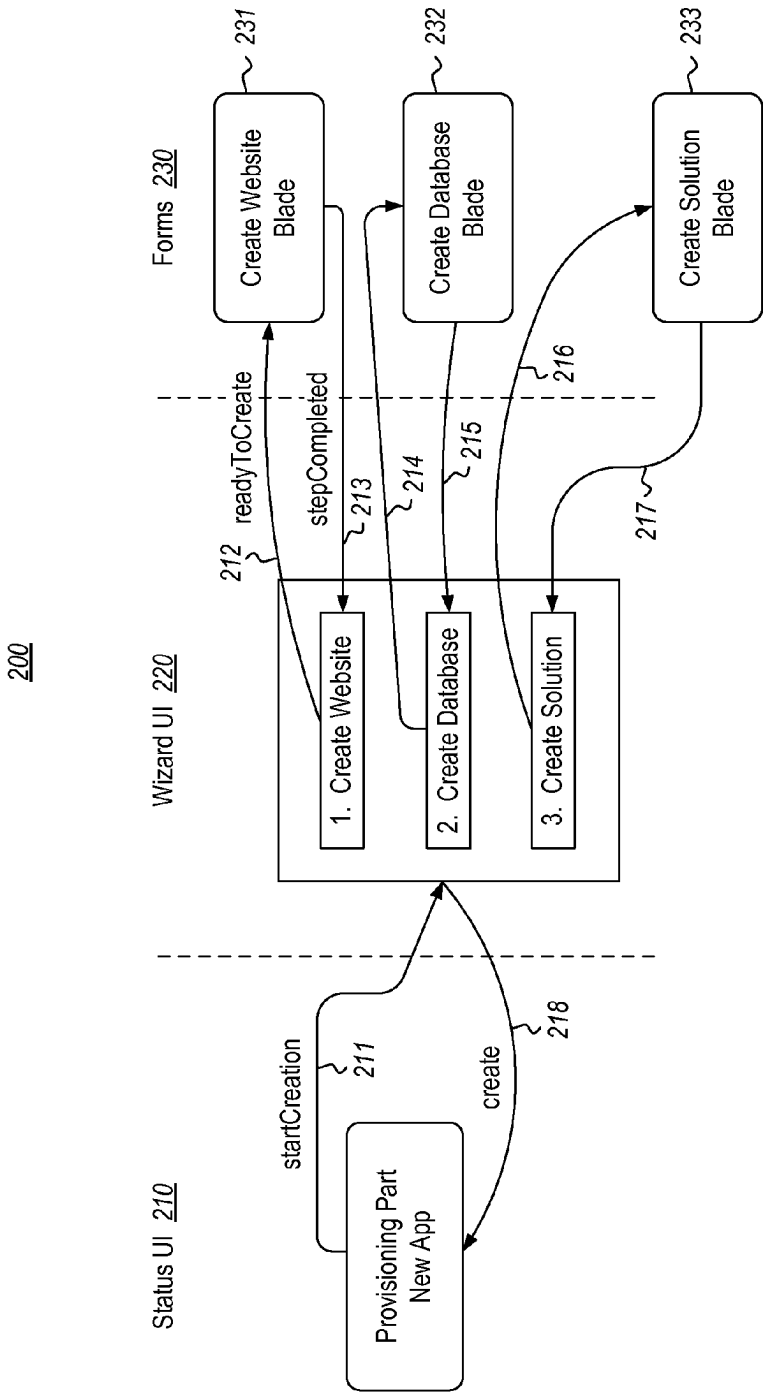


Figure 2

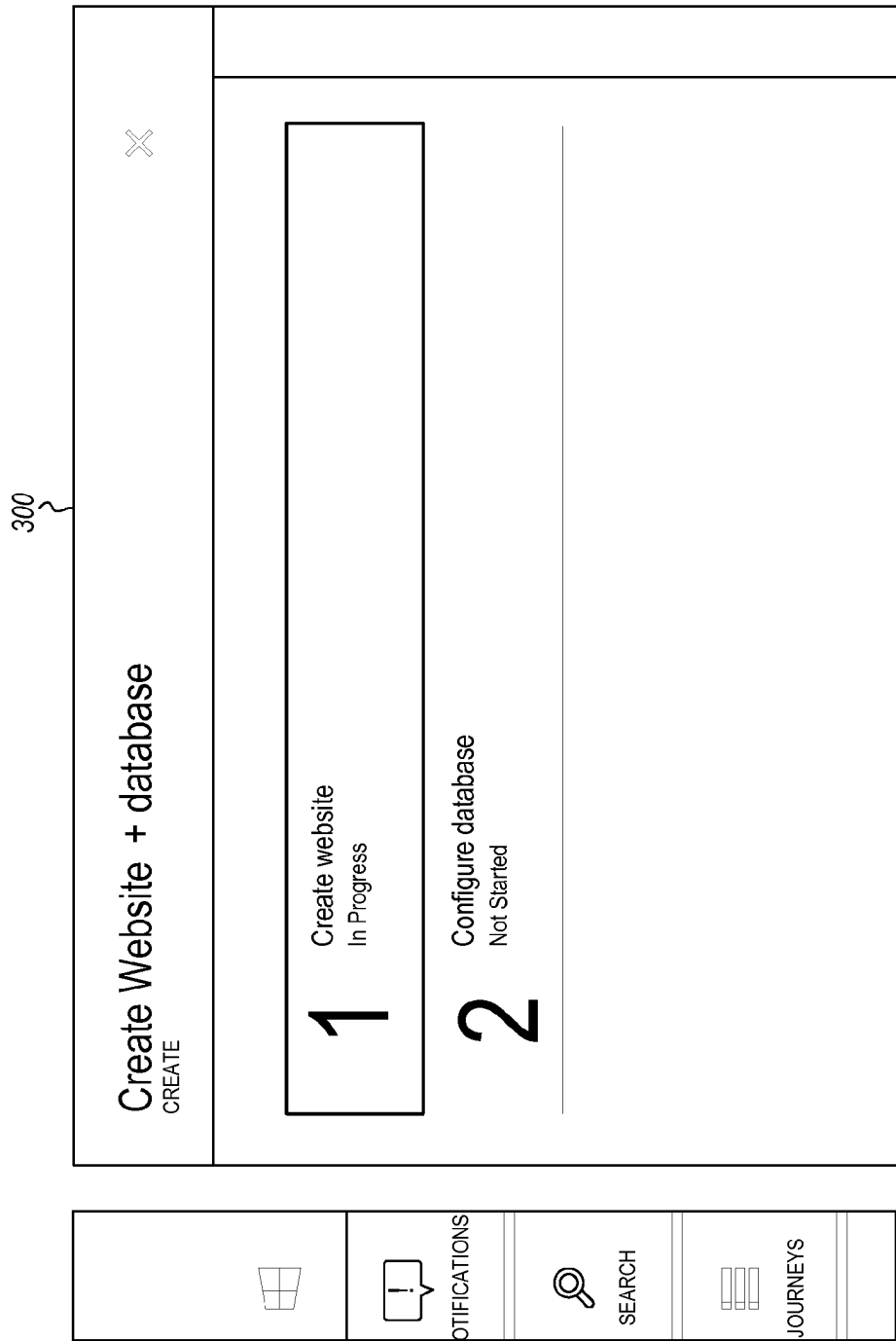


Figure 3

Create Website + database

CREATE

✓

Create website
Finished

2

Configure database
In Progress

Figure 4

Create Website

CREATE WEBSITE

URL

foo

.azurewebsites.net

REGION

West Europe

MODE ⓘ

FREE

SHARED

STANDARD

Next

Figure 5

Create Website
CREATE WEBSITE

URL

foo

!

region

West Europe

v

.azurewebsites.net

MODE ⓘ

FREE

SHARED

STANDARD

Next

This button is disabled because of the validation error

The web site name foo is not available

Figure 6

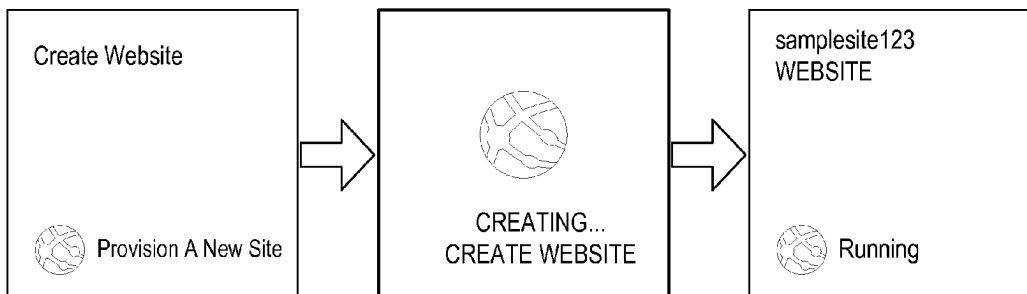


Figure 7

MULTI-ACT CREATION USER INTERFACE ELEMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of each of the following provisional patent applications, and each of the following provisional patent applications are incorporated herein by reference in their entirety:

- [0002]** 1. U.S. Provisional Application Ser. No. 61/905, 129, filed Nov. 15, 2013;
- [0003]** 2. U.S. Provisional Application Ser. No. 61/884, 743, filed Sep. 30, 2013;
- [0004]** 3. U.S. Provisional Application Ser. No. 61/905, 111, filed Nov. 15, 2013;
- [0005]** 4. U.S. Provisional Application Ser. No. 61/905, 243, filed Nov. 17, 2013;
- [0006]** 5. U.S. Provisional Application Ser. No. 61/905, 114, filed Nov. 15, 2013;
- [0007]** 6. U.S. Provisional Application Ser. No. 61/905, 116, filed Nov. 15, 2013;
- [0008]** 7. U.S. Provisional Application Ser. No. 61/905, 105, filed Nov. 15, 2013;
- [0009]** 8. U.S. Provisional Application Ser. No. 61/905, 247, filed Nov. 17, 2013;
- [0010]** 9. U.S. Provisional Application Ser. No. 61/905, 101, filed Nov. 15, 2013;
- [0011]** 10. U.S. Provisional Application Ser. No. 61/905, 128, filed Nov. 15, 2013; and
- [0012]** 11. U.S. Provisional Application Ser. No. 61/905, 119, filed Nov. 15, 2013.

BACKGROUND

[0013] Computing systems allow a wide variety of tasks to be performed, some simple and some complex, and some requiring interaction with the user. When the user is involved in a computing task that has multiple phases, the computing system often presents the user with a multi-step user interface element, often called a “wizard”. The wizard guides the user through the task one step at a time, so that the user is guided through the task one step at a time.

[0014] The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one exemplary technology area where some embodiments described herein may be practiced.

SUMMARY

[0015] At least some embodiments described herein relate to a multi-act creation user interface element that is displayed when a user instructs a computing system to initiate creation of a resource that has multiple associated creation processing acts. For each of some or all of the creation processing acts, the multi-act creation user interface element causes a set of one or more controls to appear, which the user may then enter configuration information into. The multi-act creation user interface element persists creation information so entered so that the entered creation information is available for further creation processing acts.

[0016] 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.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0018] FIG. 1 abstractly illustrates an example computing system in which the principles described herein may be employed;

[0019] FIG. 2 illustrates an architecture and process in which there are three tiers of user interface element; namely, a status user interface element, a wizard user interface element (or the “multi-act creation user interface element”), and the respective creation processing act forms;

[0020] FIG. 3 illustrates a wizard user interface element involved with creating a resource that includes a website and an associated configured database;

[0021] FIG. 4 illustrates the wizard user interface element of FIG. 3, except that the wizard user interface element now shows that the creation processing act of creating the website is finished;

[0022] FIG. 5 illustrates an example form involved with the “Create website” creation processing act of FIGS. 3 and 4;

[0023] FIG. 6 illustrates an example form that is similar to that of FIG. 5, except with a validation error shown; and

[0024] FIG. 7 illustrates a process flow wherein a user interface element is morphed from a user interface element that may be selected to create a resource, to a provisioning user interface element showing that the resource is being created, to a resource completed user interface element that may be selected to access the created resource.

DETAILED DESCRIPTION

[0025] In accordance with embodiments described herein, a multi-act creation user interface element (also called herein a “wizard”) is described. The wizard user interface element is displayed when a user instructs a computing system to initiate creation of a resource that has multiple associated creation processing acts. For each of some or all of the creation processing acts, the multi-act creation user interface element causes a set of one or more controls to appear, perhaps as a form, which the user may then enter configuration information into. The multi-act creation user interface element persists creation information so entered so that the entered creation information is available for further creation processing acts. Each of one or more of the creation processing acts displays a form having one or more controls that the user enters configuration information into. The creation processing acts may be facilitated by different applications while still being guided by the wizard user interface element.

[0026] The principles described herein may be implemented using a computing system. For instance, the users may be engaging with the system using a client computing

system. The executable logic supporting the system and providing visualizations thereon may also be performed using a computing system. The computing system may even be distributed. Accordingly, a brief description of a computing system will now be provided.

[0027] Computing systems are now increasingly taking a wide variety of forms. Computing systems may, for example, be handheld devices, appliances, laptop computers, desktop computers, mainframes, distributed computing systems, or even devices that have not conventionally been considered a computing system. In this description and in the claims, the term “computing system” is defined broadly as including any device or system (or combination thereof) that includes at least one physical and tangible processor, and a physical and tangible memory capable of having thereon computer-executable instructions that may be executed by the processor. The memory may take any form and may depend on the nature and form of the computing system. A computing system may be distributed over a network environment and may include multiple constituent computing systems. An example computing system is illustrated in FIG. 1.

[0028] As illustrated in FIG. 1, in its most basic configuration, a computing system **100** typically includes at least one processing unit **102** and memory **104**. The memory **104** may be physical system memory, which may be volatile, non-volatile, or some combination of the two. The term “memory” may also be used herein to refer to non-volatile mass storage such as physical storage media. If the computing system is distributed, the processing, memory and/or storage capability may be distributed as well. As used herein, the term “executable module” or “executable component” can refer to software objects, routines, or methods that may be executed on the computing system. The different components, modules, engines, and services described herein may be implemented as objects or processes that execute on the computing system (e.g., as separate threads).

[0029] In the description that follows, embodiments are described with reference to acts that are performed by one or more computing systems. If such acts are implemented in software, one or more processors of the associated computing system that performs the act direct the operation of the computing system in response to having executed computer-executable instructions. For example, such computer-executable instructions may be embodied on one or more computer-readable media that form a computer program product. An example of such an operation involves the manipulation of data. The computer-executable instructions (and the manipulated data) may be stored in the memory **104** of the computing system **100**. Computing system **100** may also contain communication channels **108** that allow the computing system **100** to communicate with other message processors over, for example, network **110**.

[0030] The computing system **100** also includes a display **112** on which a user interface, such as the user interfaces described herein, may be rendered. Such user interfaces may be generated in computer hardware or other computer-represented form prior to rendering. The presentation and/or rendering of such user interfaces may be performed by the computing system **100** by having the processing unit(s) **102** execute one or more computer-executable instructions that are embodied on one or more computer-readable media. Such computer-readable media may form all or a part of a computer program product.

[0031] Embodiments described herein may comprise or utilize a special purpose or general-purpose computer including computer hardware, such as, for example, one or more processors and system memory, as discussed in greater detail below. Embodiments described herein also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are physical storage media. Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the invention can comprise at least two distinctly different kinds of computer-readable media: computer storage media and transmission media.

[0032] Computer storage media includes RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other tangible medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0033] A “network” is defined as one or more data links that enable the transport of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmission media can include a network and/or data links which can be used to carry or desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

[0034] Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to computer storage media (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile computer storage media at a computer system. Thus, it should be understood that computer storage media can be included in computer system components that also (or even primarily) utilize transmission media.

[0035] Computer-executable instructions comprise, for example, instructions and data which, when executed at a processor, cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to

the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0036] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, routers, switches, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0037] As will be described herein system presents a user interface on a display (such as the display 112 of FIG. 1). The system may have one or more applications running therein. The system may be thought of as an operating system, in which case the one or more applications are simply applications running in the operating system.

[0038] On the other hand, the system may also itself be just an application (such as a web browser), in which case the applications that run within the system may be thought of as “extensions” to the application. In the description and in the claims, such an “extension” will be termed an “application”.

[0039] In accordance with the principles described herein, a user may interface with the portal to create resources, even if the contribution of multiple entities are used to create the resource. For instance, one creation processing act may involve the system itself, whereas another creation processing act may involve an application running in that system. As another example, one creation processing act may involve a first application, and another creation processing act may involve another application. The portal provides a multi-act creation user interface element (also called herein a “wizard” or “wizard user interface element”) that allows the user to create complex resources in a single create operation, potentially with components provided by different entities. This is a complex problem as the creation of resources in distributed systems often involves client and server side interactions of code from multiple applications, and orchestration across creation processing acts involved in the creation of the resource.

[0040] In accordance with embodiments described herein, there is provided a way to govern the set of creation processing acts involved in creating a resource, even for long running execution. Furthermore, input is acquired and contemporaneously validated to ensure correctness. The system may allow sharing of data between different applications that contribute to the creation, and provide a consistent experience across different applications. The creation data involved with creating the resource may be stored, and even shared between different elements in the user interface that contribute to the creation.

[0041] The portal displays and collects data across multiple components that can come from different applications (e.g., different web sites). In order to collect that information, a full experience is provided that enables application developers to describe the logic required for creating one or related set of

assets with a given order and conformance rules. To enable these set of scenarios, a number of capabilities are provided.

[0042] First, the portal projects forms, perhaps one for each of the multiple creation processing acts involved in the creation of the resource. Each form has a set of one or more controls that collect creation data. Such creation data may be input into the respective control by the user, and perhaps may have a default value that was perhaps received from creation data inputted into a prior creation processing act.

[0043] Second, each of the one or more controls in each form may contain validation rules that determine if the input is valid or not. These validation rules may be built-in (and invoked via declarative composition) or provided as code by the application developer. The system will determine where that validation code will run (in the main system or in the application’s execution space) to ensure best responsiveness of the system.

[0044] Third, validations and forms may be described declaratively (e.g., as a view model and a template). All the interactions are provided by the underlying executable controls. Moreover, the application developer has control over the layout but the controls provide the lower-level interaction.

[0045] Fourth, a multi-act resource creation element (also referred to as a “wizard” herein) describes a set of creation processing acts that are to be done in a given order. Each creation processing act collects information and that information is collected using one or more controls (such as in a form).

[0046] Fifth, the wizard user interface element has rules to govern what creation processing act (i.e., what form) is displayed and if any new action needs to be taken dynamically. In addition, the creation processing acts can come from different applications or extensions (e.g., web sites) as the applications can contribute different user interface elements.

[0047] Sixth, the data from all the forms is collected and persisted by the system in space (hereinafter the space being called an “EditScope” or “edit scope”). The space is memory space that collects data and tracks changes in entities. This edit scope can survive system refreshes and can be long-running. For instance, if the system is an operating system, the edit scope survives restart of the operating system. If the system is a browser, the edit scope survives browser refresh.

[0048] Seventh, when creating a resource (which may be a complex multi-act operation), a basic user interface element (called herein a “processing status user interface element” or “provisioning user interface element”) is placed in any part of the portal and represents the resource being created. Once the wizard user interface element is completed, the wizard signals back to the provisioning user interface element to execute the actual creation command. If the creation command is executed successfully, the provisioning user interface element is replaced with a resource created status user interface element, which may be a user interface element that represents the created resource itself.

[0049] Eighth, communication between all these user interface elements may be through a given protocol (e.g., by sending edit scope and metadata forward and receiving signals back upon completion).

[0050] Ninth, sometimes the wizard can be used without imposing any execution order. For example, the wizard can be displayed to a user to suggest a set of tasks that need to be performed in no particular order

[0051] Embodiments described herein may address some or all the challenges associated with the creation of simple and complex resources.

[0052] FIG. 2 illustrates a process 200 in which there are three tiers of user interface element, a status user interface element 210, a wizard user interface element 220 (or the “multi-act creation user interface element”), and the respective creation processing act forms 230. In the illustrated example of FIG. 2, there are three forms 231, 232 and 233, associated with three creation processing acts.

[0053] Initially, the user instructs the computing system to initiate creation of a resource. The principles described herein are not limited to how the user does this. However, in one embodiment, there is a resource creation initialization user interface element that gives the user the option of selecting the user interface element to thereby instruct the system to create the corresponding resource. In response, the system may replace (or morph into) the resource creation initialization user interface element with a provisioning user interface element, which is represented by the status user interface element 210 of FIG. 2.

[0054] The provisioning part 210 then creates the edit scope, which will be used to persist creation data as the user inputs that creation data into various forms during creation. If the resource being created is a complex resource (i.e., involving multiple creation processing acts), the provisioning parts 210 signals (see arrow 211) for the multiple-act creation (i.e., “wizard”) user interface element 220 to be created.

[0055] The wizard user interface element 220 has knowledge of all the necessary creation processing acts (which may be represented by forms) to create the resource. For instance, in this example, there are three sequential creation processing acts to be performed; namely, 1) create website, 2) create database, and 3) create solution. The wizard user interface element 220 understands dependencies between each creation processing act, keeps track of the status of each creation processing act, and determines when progress has proceeded to a point where another not-yet-started creation processing act may begin. In the case of sequential creation processing acts, the wizard user interface element 220 simply proceeds one creation processing act at a time.

[0056] When the wizard user interface element 220 determines that the new creation processing act is to be initiated, the wizard user interface element 220 causes the corresponding form to be created. For instance, in FIG. 2, as represented by arrow 212, the wizard user interface element 220 signals for creation of the form 231, which may be used to create a web site. The form 231 receives perhaps some of the edit scope (e.g., web site name) to allow for population of some of the controls and/or conditional presentation of some or all of the controls. For instance, the form 231 may be mapped to a view model which exposes the edit scope to populate fields of the form 231. The form 231 also collects information from the user, and the various controls of the form also validate the information as it is input into the control. The form 231 also has a button or control that the user may interact with to indicate completion of the creation processing act. Once complete, the creation processing act signals 231 back to the wizard user interface element 220 (as represented by the arrow 213), representing completion of the creation processing act. In this process, the input creation data from form 231 is persisted to the edit scope. Optionally, the form 231 may then be caused to disappear.

[0057] The wizard user interface element 220 responds to each completion signal by either determining that the entire creation is completed, or identifying the next creation processing act (in this case act 232). For instance, in FIG. 2, as represented by arrow 214, the wizard user interface element 220 signals for creation of the second form 232, which is used to create a database. The form 232 receives perhaps some of the edit scope to allow for population (e.g., via an appropriate view model) of some of the controls and/or conditional presentation of some or all of the controls. The form 232 also collects information from the user, and the various controls of the form also validate the information as it is input into the control. The form 232 also has a button or control that the user may interact with to indicate completion of the creation processing act. Once complete, the creation processing act 232 signals back to the wizard user interface element 220 (as represented by the arrow 215), representing completion of the creation processing act 232. In this process, the input creation data from form 232 is persisted to the edit scope. Optionally, the form 232 may then be caused to disappear.

[0058] The wizard user interface element 220 then identifies the next creation processing act (in this case act 233). For instance, in FIG. 2, as represented by arrow 216, the wizard user interface element 220 signals for creation of the second form 233, which is used to create a solution using the created website and database. The form 233 receives perhaps some of the edit scope to allow for population of some of the controls and/or conditional presentation of some or all of the controls. The form 233 also collects information from the user, and the various controls of the form also validate the information as it is input into the control. The form 233 also has a button or control that the user may interact with to indicate completion of the creation processing act. Once complete, the creation processing act 233 signals back to the wizard user interface element 220 (as represented by the arrow 217), representing completion of the creation processing act 233. In this process, the input creation data from form 233 is persisted to the edit scope. Optionally, the form 233 may then be caused to disappear.

[0059] Here, the wizard user interface element 220 determines that there are no further creation processing acts to occur. Thus, the wizard user interface element 220 signals to the provisioning user interface element 210 (as represented by arrow 218) that the resource creation is completed. The provisioning user interface element may then be replaced by a resource created status user interface element, which is a representation of the created resource. In the future, when the resource is to be used, the user may simply select the resource created status user interface element.

[0060] From this example, it can be seen that as every creation processing act is completed, the corresponding form signals back to the wizard user interface element indicating that the creation processing act is done. Once the wizard user interface element receives the signal, the wizard user interface element moves to the next creation processing act. If all the creation processing acts are completed, the wizard user interface element signals back to the provisioning user interface element. The provisioning user interface element executes the create command and mutates to the created resource.

[0061] If the resource is created in just one act, the signal form is shown without the use of the wizard user interface element. Once the form is valid and completed, the form

signals back to the provisioning part. The provisioning part then executes the command and mutates to the created resource.

[0062] The forms may be composable. For instance, the create website form **231** may be used as a first creation processing act component in the creation of a compound solution, as shown in FIG. 2. However, the create website form may also be used in a single act creation, in which the user merely desires to create a web site. Thus, the forms are composable, and can be used in different create operations. This is because in each create operation, the form is able to receive the appropriate edit scope.

[0063] Each form includes a button or control that the user may interact with in order to indicate that the corresponding creation processing act is complete. In some embodiments, the control is not enabled if creation data input into the form has not been successfully validated.

[0064] The wizard user interface element allows application developers to define a set of creation processing acts that result in the creation of a set of resources. For instance, FIG. 3 illustrates a wizard user interface element **300** involved with creating a resource that includes a website and an associated configured data base.

[0065] Creation processing acts can be optional. The wizard user interface element is aware of when creation processing acts are completed and shows a visual indication of the completion. For instance, FIG. 4 is similar to FIG. 3, except that the wizard user interface element now shows the step of creating the website is finished.

[0066] In addition, the wizard user interface element has access to collected data (via the edit scope) in all creation processing acts and can perform validations across different creation processing acts from different applications. For instance, if the user tried to create a website and a database in different regions of the world, the user might get a warning about latency. Although conventional wizards imply a certain order of execution, the wizard user interface element described herein supports sets of creation processing acts without order, and also supports optional steps.

[0067] In some embodiments, each creation processing act in the wizard user interface element displays a form. For instance, FIG. 5 illustrates an example form involved with the “Create website” creation process act of FIGS. 3 and 4.

[0068] The form is created using a forms infrastructure from the portal that leverages the available portal controls. Application developers may define the form fields in their view models, and the form layout in the template. Thus, application developers may have control over the layout of the form. Each field has a type that determines the control to be used to acquire the data. In addition, fields can be associated with validations. The form is aware of the validation status and does not allow a creation processing act to complete if the validations do not pass. For instance, FIG. 6 illustrates an example form that is similar to that of FIG. 5, except with a validation error shown.

[0069] Also validations have a consistent user experience and behavior that is provided by the underlying controls. Forms provide complex functionality like creation of underlying controls to acquire values, dirty field tracking, and validations.

[0070] There is a set of available validations that the user can use. These validations may be declaratively associated with a field. In addition, application developers can write asynchronous validators that run their custom code (within

our isolation model in which the execution is within an application execution environment). An example of this is the validation that checks that a website name is not taken.

[0071] As previously mentioned, the create data created from a create experience is store in a container called “edit scope” or “editScope”. The edit scope is a data container that contains a set of data along with differential changes made to it since its creation. Forms and edit scope are very tightly integrated. Each form field projects a property of an object in an edit scope. All these properties are observable, so changes are propagated across the stack. For instance, an input in a field results in changes in the underlying edit scope without the need of developer code. As a result of this, the edit scope provides access to all the data collected during the wizard/create operation plus the ability to know if the data is dirty. Regarding the edit scope, in general terms, forms go through the following process: they acquire the edit scope (by dereferencing it via an editScope identifier that is passed via a binding) and map it to the relevant fields in the form.

[0072] The life-cycle of the edit scope is controlled by the system and it can be specified declaratively. It can be associated with a user interface element (and hence propagated to all its children). The way in which edit scope is shared between user interface elements is by passing the edit scope via regular portal bindings. EditScopes may be automatically persisted to the cloud. An implication of this is that create operations can be long running (e.g. a complex wizard with multiple steps can be completed in multiple days).

[0073] There may be an intrinsic user interface element called “provisioning” that represents a resource that is being created. This provisioning user interface element may be placed in any part of the portal when a create operation starts. It owns the edit scope associated with the creation (which will be passed to all children that will contribute to that editScope). If the creation process has more than one creation processing act, the wizard user interface element will be displayed (when the user interface element is selected). Otherwise, the portal will just show the form to collect the data. The provisioning user interface element also invokes the command to perform the actual provisioning, so it waits for a completion signal from the wizard user interface element (in the case of multi-act creation) or the form (in the case of single act creation). Once the provisioning user interface element receives the completion signal, the provisioning user interface element executes the command. If there are no errors, the provisioning user interface element mutates to the resource that has been created as shown in FIG. 7.

[0074] FIG. 7 illustrates a process flow wherein a user interface element is morphed from a user interface element that may be selected to create a resource, to a provisioning user interface element showing that the resources is being created, to a resource completed user interface element that may be selected to access the created resource.

[0075] In one embodiment, the portal may include a canvas that extends in an extendable direction (i.e., rightward). User interface elements called blades may be added to the canvas, and are added to the end of a sequence of one or more blades that may already exist on the canvas. The canvas is extendible in the extendible direction in the sense that as new blades are added, they are added to the sequence to extend the sequence at the extendible direction (e.g., rightward). The blades may occupy substantially all of the extent of the canvas within a given range of the extendible dimension (e.g., horizontal), and occupy substantially all of the canvas in the direction

orthogonal to the extendible dimension. In this context, as creation of a resource requiring multiple acts is initiated, the wizard user interface element may be added as a blade to the extendible canvas. As the user engages in particular resource creation acts, the corresponding form may additionally be added in the extendible dimension, and then perhaps disappear when that creation processing act is completed.

[0076] Accordingly, the principles described herein provide an effective mechanism to allow users to create resources in composable systems, in which multiple applications contributed to the creation.

[0077] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed is:

1. A computer program product comprising one or more computer-readable storage media having thereon computer-executable instructions that, when executed by one or more processors of a computing system, cause the computing system to instantiate and/or operate the following:

a multi-act resource creation element configured to appear when a user instructs the computing system to initiate creation of a resource, each of the at least some of the plurality of creation processing acts involving inputting of creation information that the set of one or more controls will use to create the resource, the multi-act resource creation element persisting creation information entered in each of the at least some of the plurality of creation processing acts.

2. The computer program product in accordance with claim 1, the plurality of creation processing acts including a first creation processing act and a second creation processing act, the multi-act resource creation element including a plurality of indicators each associated with a corresponding act of a plurality of creation processing acts involved in creating the resource,

the first creation processing act corresponding to a first indicator of the plurality of indicators so that the first indicator is visually emphasized during the first creation processing act; and

the second creation processing act corresponding to a second indicator of the plurality of indicators so that the second indicator is visually emphasized during the second creation processing act.

3. The computer program product in accordance with claim 1, the multi-act resource creation element configured to 1) cause a first set of one or more creation input controls to be displayed during a first creation processing act of the plurality of creation processing acts, and 2) save configuration inputted into the first set of one or more creation input controls in conjunction with the first creation processing act completing.

4. The computer program product in accordance with claim 3, the multi-act resource creation element configured to 1) cause a second set of one or more creation input controls to be displayed during a second creation processing act of the plurality of creation processing acts, and 2) save configuration inputted into the second set of one or more creation input controls in conjunction with the second creation processing act completing.

5. The computer program product in accordance with claim 4, the first set of one or more creation input controls provided by a first application running in a system, and the second set of one or more creation input controls provided by a second application running in the system.

6. The computer program product in accordance with claim 4, the first set of one or more creation input controls provided by a system, and the second set of one or more creation input controls provided by an application running in the system.

7. The computer program product in accordance with claim 4, the one or more computer-readable storage media further having thereon computer-executable instructions that, when executed by the one or more processors of the computing system, cause the computing system to instantiate and/or operate the following:

the first set of one or more creation input controls displayed as a first form; and

the second set of one or more creation input controls displayed as a second form.

8. The computer program product in accordance with claim 7, the first set of one or more creation input controls provided by a first application running in a system, and the second set of one or more creation input controls provided by a second application running in the system or the system itself.

9. The computer program product in accordance with claim 7,

the first set of one or more creation input controls performing validation checking based on a first set of persisted validation rules, and

the second set of one or more creation input controls performing validation checking based on a second set of persisted validation rules.

10. The computer program product in accordance with claim 3, the one or more computer-readable storage media further having thereon computer-executable instructions that, when executed by the one or more processors of the computing system, cause the computing system to instantiate and/or operate the following:

the first set of one or more creation input controls displayed as a first form, and configured to performing validation checking based on a first set of persisted validation rules.

11. The computer program product in accordance with claim 10, the first set of one or more creation input controls further performing the first resource creation processing act upon successful validation checking of the first form.

12. The computer program product in accordance with claim 10, the multi-act resource creation element being added to an extendable canvas in an extendible direction of the canvas, the first form being added to the extendable canvas adjacent the multi-step resource creation element in the extendible direction of the canvas when the first creation processing act is initiated, and the first form being removed from the extendable canvas when the first creation processing act is completed.

13. A method for using a user interface of a computing system to guide a user through a multi-act process for creating a resource, the method comprising:

an act of detecting that the user has instructed the computing system to initiate creation of a resource;

in response to the instruction, an act of causing a multi-act resource creation element to appear on a display of the computing system, the computing system performing the following for each of at least some of the plurality of creation processing acts:

an act of causing a corresponding set of one or more controls to be displayed; and

an act of persisting creation input entered by the user into the corresponding set of one or more controls.

14. The method in accordance with claim **13**, the multi-act resource creation element including a plurality of indicators each associated with a corresponding act of a plurality of creation processing acts involved in creating the resource.

15. The method in accordance with claim **13**, the act of persisting for a given one of the creation processing acts occurring prior to a display of a next set of one or more controls associated with a next creation processing act.

16. The method in accordance with claim **15**, the method also including the following for each of the at least some of the plurality of creation processing acts:

an act of the multi-act resource creation element determining that the corresponding processing act is complete based on a signal received from at least one control of the corresponding set of one or more controls.

17. The method in accordance with claim **16**, the method also including the following for each of the at least some of the plurality of creation processing acts:

an act of providing creation information that the user input in one or more prior resource processing acts to the corresponding resource processing act.

18. The method in accordance with claim **13**, further comprising the following in response to the instruction to create the resource:

an act of causing a processing status user interface element to be displayed on the display.

19. The method in accordance with claim **18**, further comprising:

an act of the computing system detecting that the resource has been successfully created; and

in response to the detecting that the resource has been successfully created, an act of causing the processing status user interface element to be replaced with a resource created status user interface element.

20. A method for using a user interface of a computing system to guide a user through a multi-act process for creating a resource, the method comprising:

an act of detecting that the user has instructed the computing system to initiate creation of a resource; and

in response to the instruction, an act of causing a multi-act resource creation element to appear on a display of the computing system, the computing system performing the following for each of at least one of the plurality of creation processing acts:

an act of providing to the corresponding resource processing act creation information that the user input in one or more prior resource processing acts of the plurality of creation processing acts;

an act of causing a corresponding set of one or more controls to be displayed as a corresponding form, wherein at least some of the creation information provided in the act of providing is manifest in the corresponding form; and

an act of persisting creation input entered by the user into the corresponding form.

* * * * *