US 20110055738A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2011/0055738 A1

Damaschke et al. (43) Pub. Date: Mar. 3, 2011

(54) CONFIGURABLE AND EXTENSIBLE USER INTERFACE

(76) Inventors: **Frank Damaschke**, Nussloch (DE); **Thomas Vogt**, Romerberg (DE)

(21) Appl. No.: **12/940,675**

(22) Filed: **Nov. 5, 2010**

Related U.S. Application Data

(63) Continuation of application No. 10/335,145, filed on Dec. 30, 2002, now Pat. No. 7,853,899.

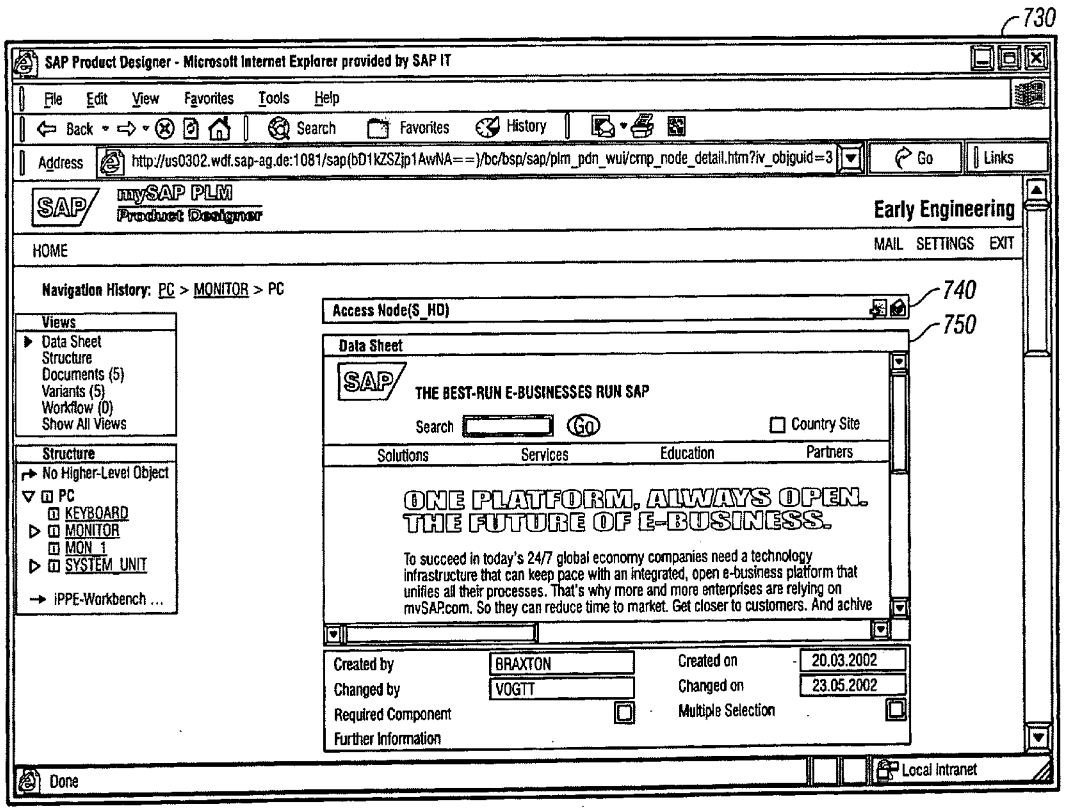
Publication Classification

(51) **Int. Cl.**
G06F 3/048 (2006.01)

(52) **U.S. Cl.** **715/763**

(57) **ABSTRACT**

Methods and apparatus, including computer program products, for changing a user interface. A method for changing a user interface layout includes presenting a user with one or more exits points into which user instructions for changes to a user interface can be inserted. The method includes receiving user instructions to change a user interface. The method includes receiving a selection of an exit point. The method includes adding the user instructions to instructions for presenting the user interface at the exit point selected.


730
SAP Product Designer - Microsoft Internet Explorer provided by SAP IT
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://us0302.wdf.sap-ag.de:1081/sap(bD1KZSZjp1AwNA==)/bc/bsp/sap/plm_pdn_wui/cmp_node_detail.htm?iv_objguid=3
Go Links
mySAP PLM
Product Designer
Early Engineering
HOME
MAIL SETTINGS EXIT
Navigation History: PC > MONITOR > PC
Views
Data Sheet
Structure
Documents (5)
Variants (5)
Workflow (0)
Show All Views
Structure
No Higher-Level Object
PC
KEYBOARD
MONITOR
MON_1
SYSTEM_UNIT
iPPE-Workbench ...
Access Node(S_HD)
740
750
Data Sheet
THE BEST-RUN E-BUSINESSES RUN SAP
Search
Go
Country Site
Solutions
Services
Education
Partners
ONE PLATFORM, ALWAYS OPEN.
THE FUTURE OF E-BUSINESS.
Done
Local intranet

To succeed in today's 24/7 global economy companies need a technology infrastructure that can keep pace with an integrated, open e-business platform that unifies all their processes. That's why more and more enterprises are relying on mySAP.com. So they can reduce time to market. Get closer to customers. And achive

Created by	BRAXTON	Created on	20.03.2002
Changed by	VOGTT	Changed on	23.05.2002
Required Component	☐	Multiple Selection	☐
Further Information			

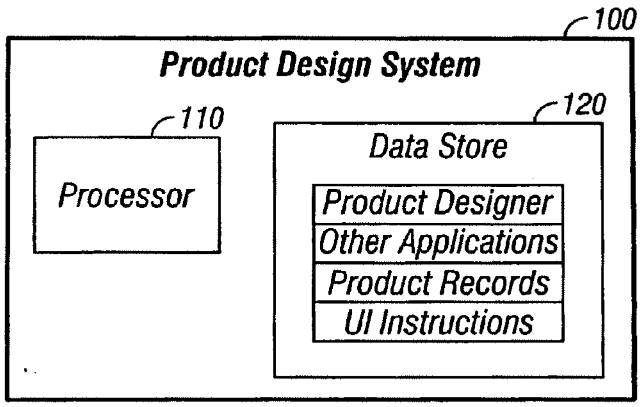

100
Product Design System
110
Processor
120
Data Store
Product Designer
Other Applications
Product Records
UI Instructions

FIG. 1

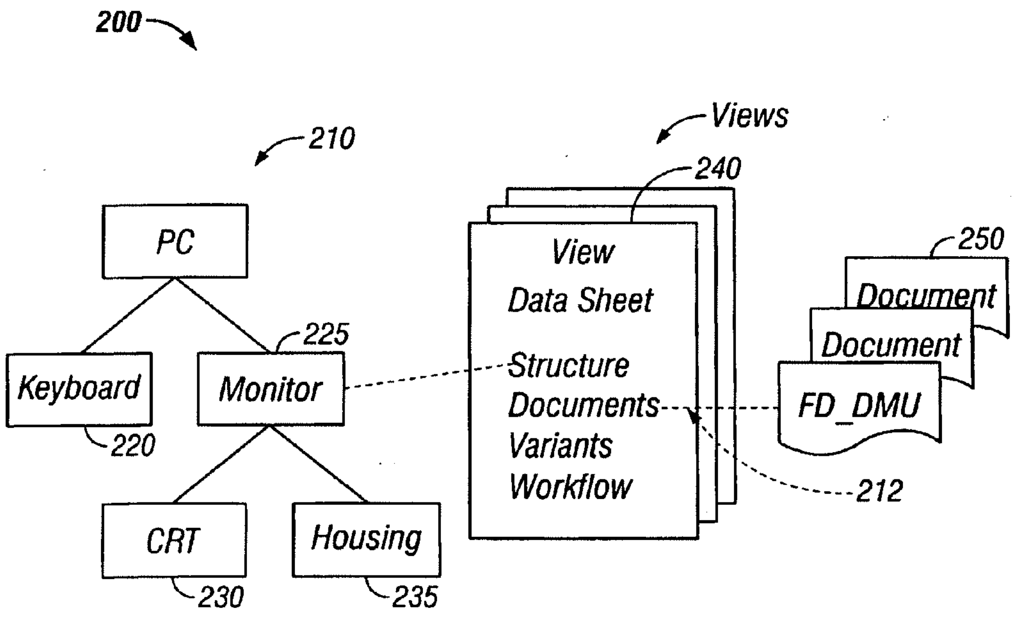
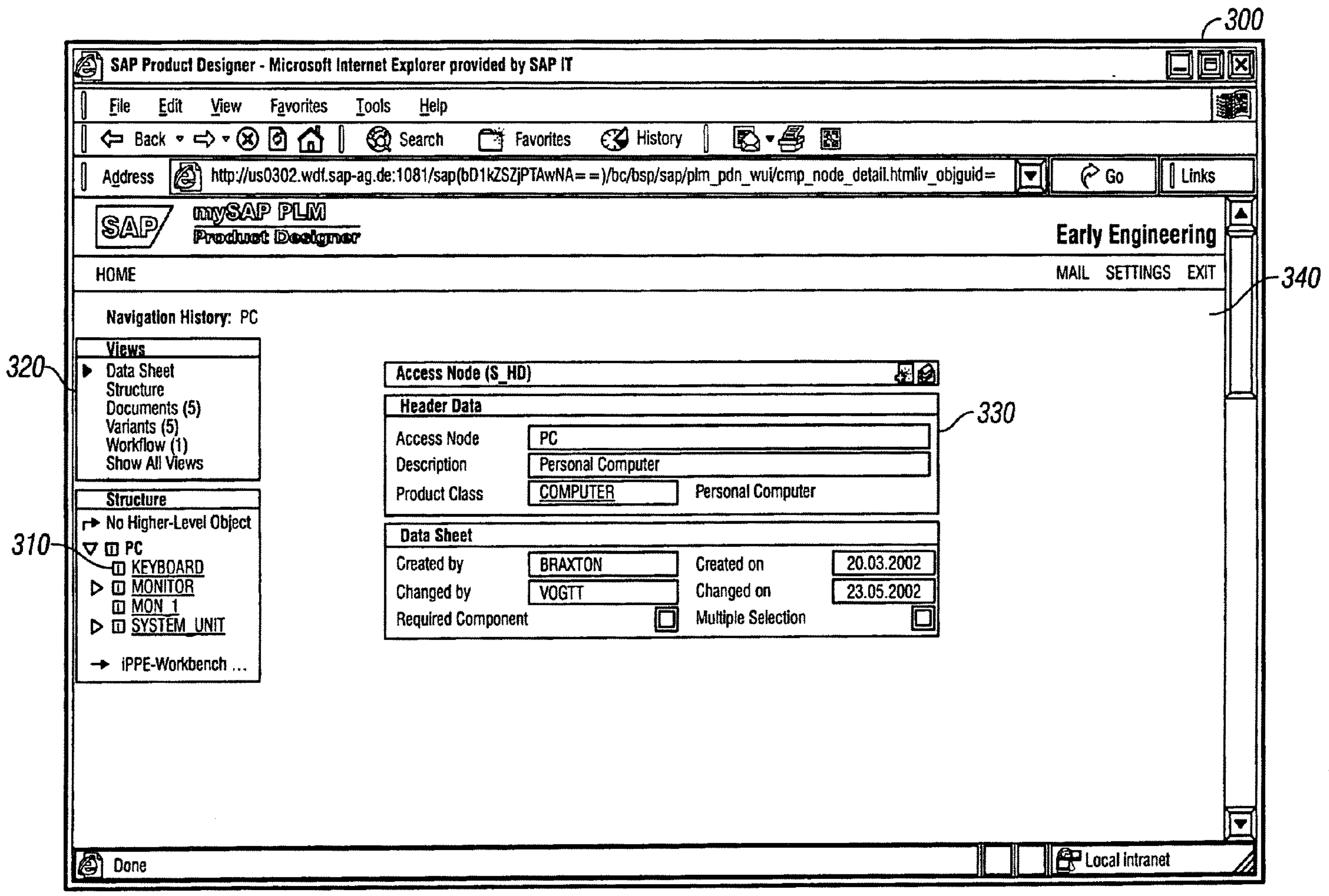

200
210
PC
Keyboard
220
Monitor
225
CRT
230
Housing
235
Views
240
View
Data Sheet
Structure
Documents
Variants
Workflow
250
Document
Document
FD_DMU
212

FIG. 2


300
SAP Product Designer - Microsoft Internet Explorer provided by SAP IT
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://us0302.wdf.sap-ag.de:1081/sap(bD1kZSZjPTAwNA==)/bc/bsp/sap/plm_pdn_wui/cmp_node_detail.htmliv_objguid=
Go Links
SAP mySAP PLM Product Designer
Early Engineering
HOME
MAIL SETTINGS EXIT
340
320
310
330
Done
Local intranet

Navigation History: PC

Views
- Data Sheet
- Structure
- Documents (5)
- Variants (5)
- Workflow (1)
- Show All Views

Structure
- No Higher-Level Object
- PC
  - KEYBOARD
  - MONITOR
  - MON_1
  - SYSTEM_UNIT
- iPPE-Workbench ...

Access Node (S_HD)

Header Data

Access Node	PC	
Description	Personal Computer	
Product Class	COMPUTER	Personal Computer

Data Sheet

Created by	BRAXTON	Created on	20.03.2002
Changed by	VOGTT	Changed on	23.05.2002
Required Component	☐	Multiple Selection	☐

FIG. 3

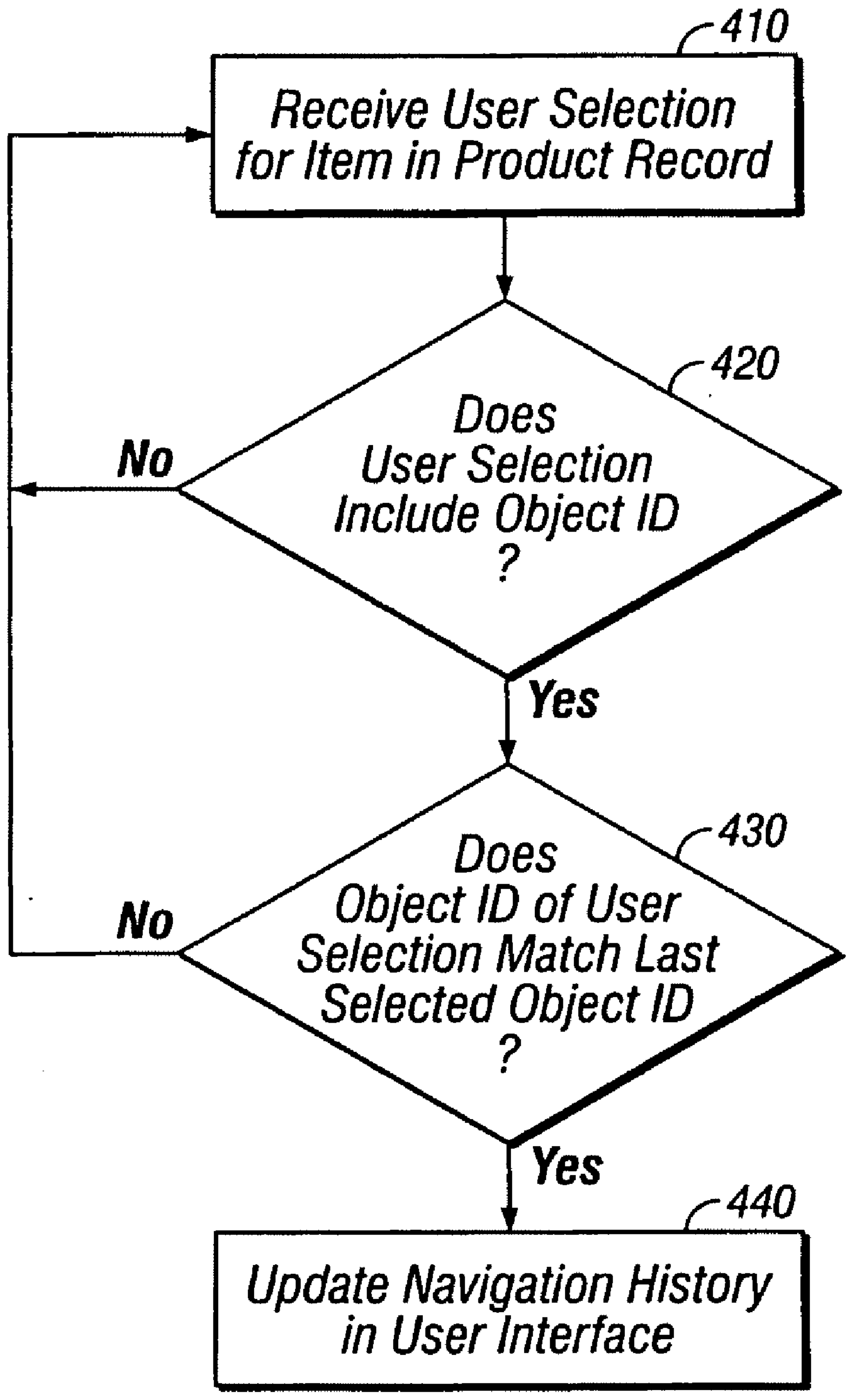
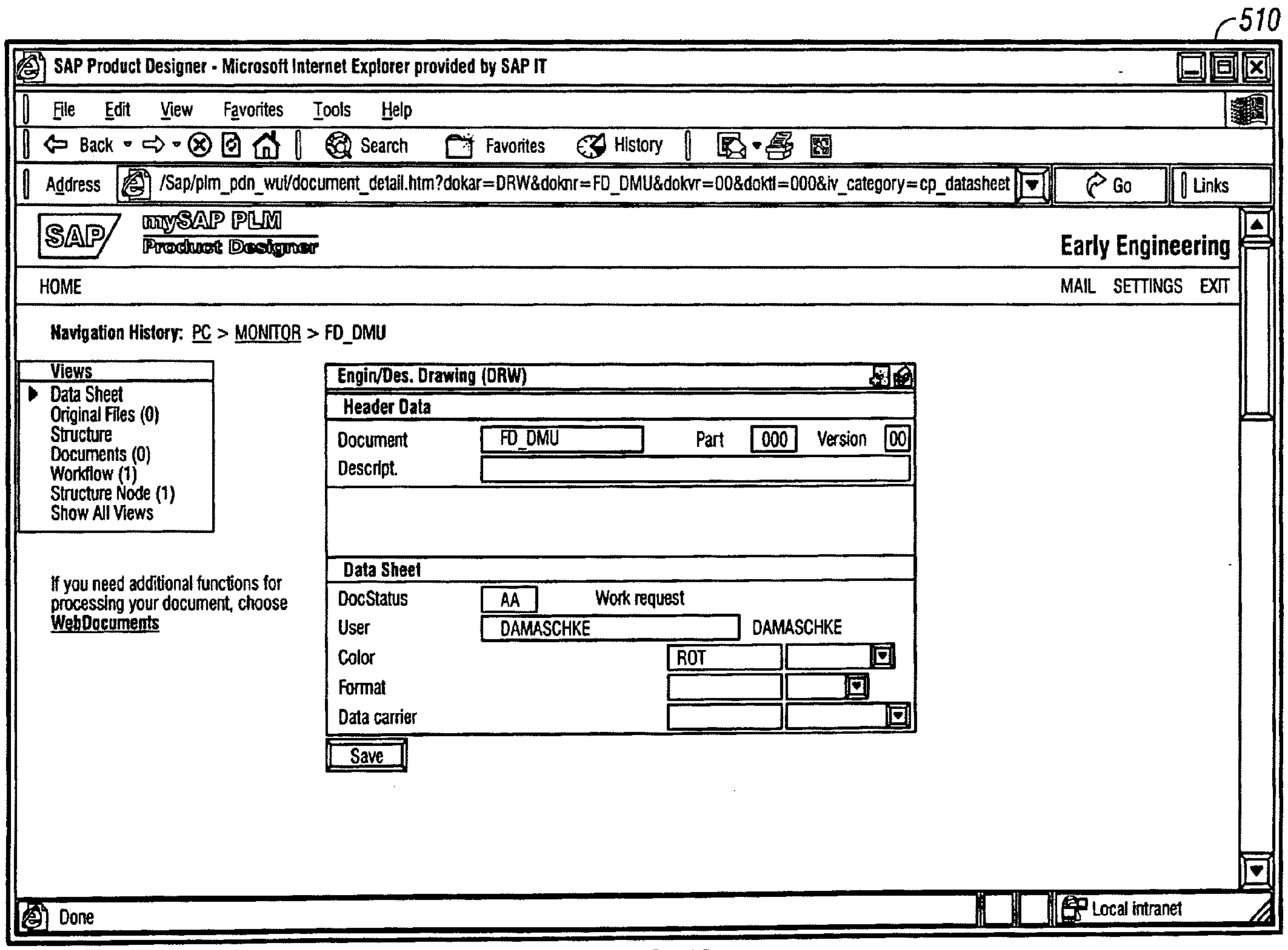

410
Receive User Selection for Item in Product Record
420
Does User Selection Include Object ID ?
No
Yes
430
Does Object ID of User Selection Match Last Selected Object ID ?
No
Yes
440
Update Navigation History in User Interface

FIG. 4


510
SAP Product Designer - Microsoft Internet Explorer provided by SAP IT
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
mySAP PLM
Product Designer
Early Engineering
HOME
MAIL SETTINGS EXIT
Views
Save
Done
Local intranet

/Sap/plm_pdn_wui/document_detail.htm?dokar=DRW&doknr=FD_DMU&dokvr=00&doktl=000&iv_category=cp_datasheet

Navigation History: PC > MONITOR > FD_DMU

Data Sheet
Original Files (0)
Structure
Documents (0)
Workflow (1)
Structure Node (1)
Show All Views

If you need additional functions for processing your document, choose WebDocuments

Engin/Des. Drawing (DRW)

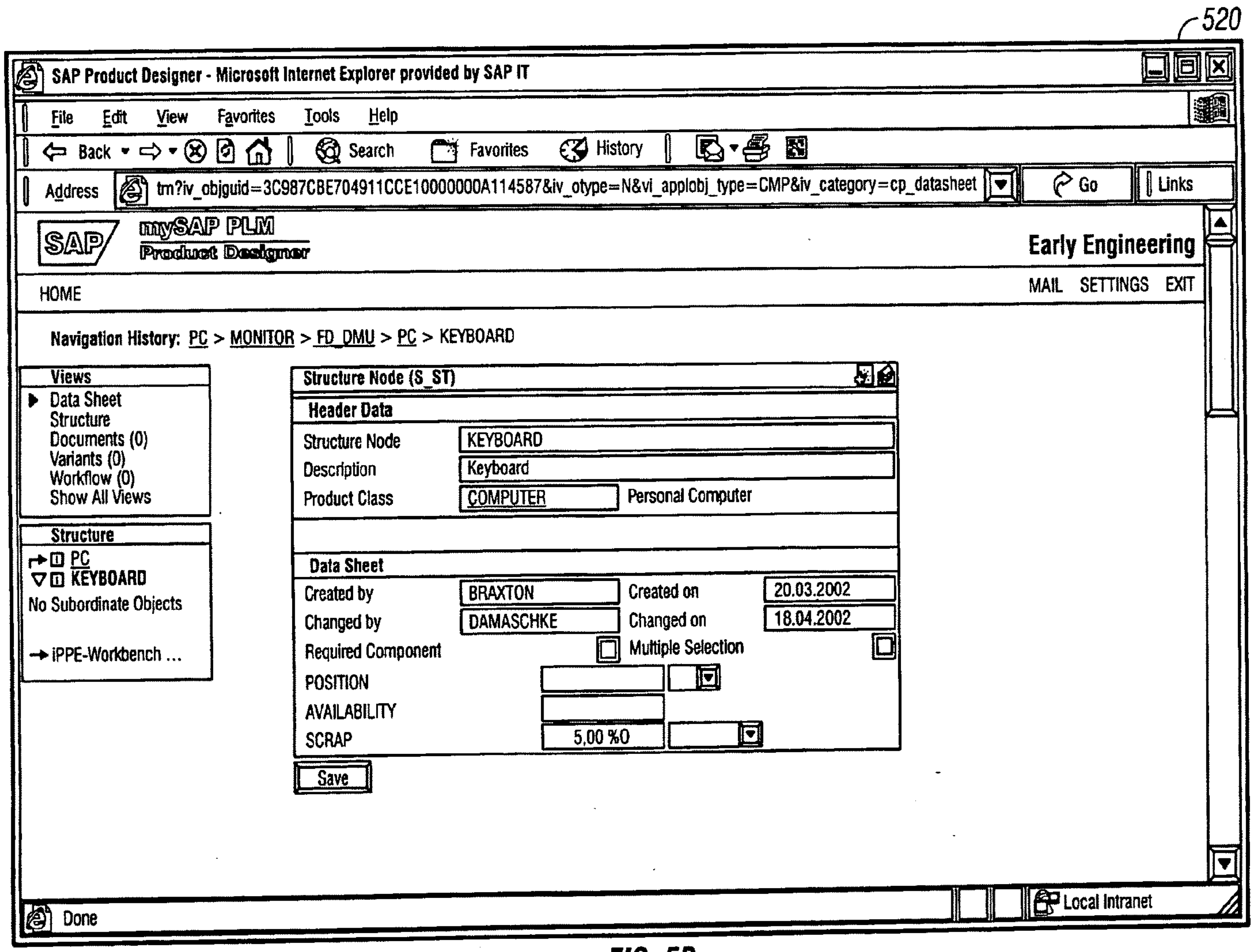
Header Data

Document FD_DMU Part 000 Version 00
Descript.

Data Sheet

DocStatus AA Work request
User DAMASCHKE DAMASCHKE
Color ROT
Format
Data carrier

FIG. 5A


520
SAP Product Designer - Microsoft Internet Explorer provided by SAP IT
File Edit View Favorites Tools Help
Back Search Favorites History
Address tm?iv_objguid=3C987CBE704911CCE10000000A114587&iv_otype=N&vi_applobj_type=CMP&iv_category=cp_datasheet
Go
Links
SAP mySAP PLM Product Designer
Early Engineering
HOME
MAIL SETTINGS EXIT
Views
Data Sheet
Structure
Documents (0)
Variants (0)
Workflow (0)
Show All Views
Structure
PC
KEYBOARD
No Subordinate Objects
iPPE-Workbench ...
Save
Done
Local Intranet

Navigation History: PC > MONITOR > FD_DMU > PC > KEYBOARD

Structure Node (S_ST)

Header Data

Structure Node	KEYBOARD		
Description	Keyboard		
Product Class	COMPUTER	Personal Computer	

Data Sheet

Created by	BRAXTON	Created on	20.03.2002
Changed by	DAMASCHKE	Changed on	18.04.2002
Required Component	☐	Multiple Selection	☐
POSITION			
AVAILABILITY			
SCRAP	5,00 %0		

FIG. 5B

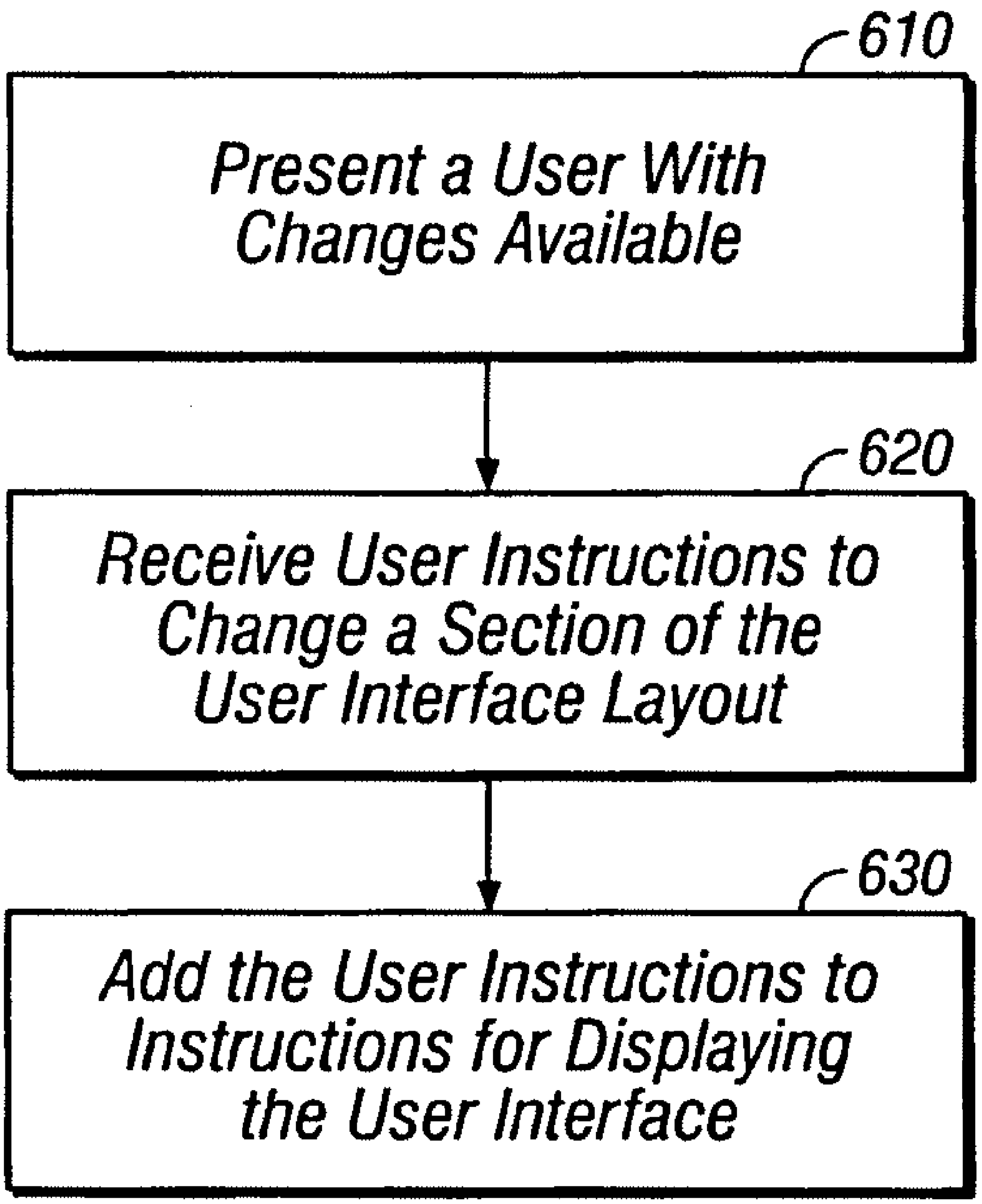
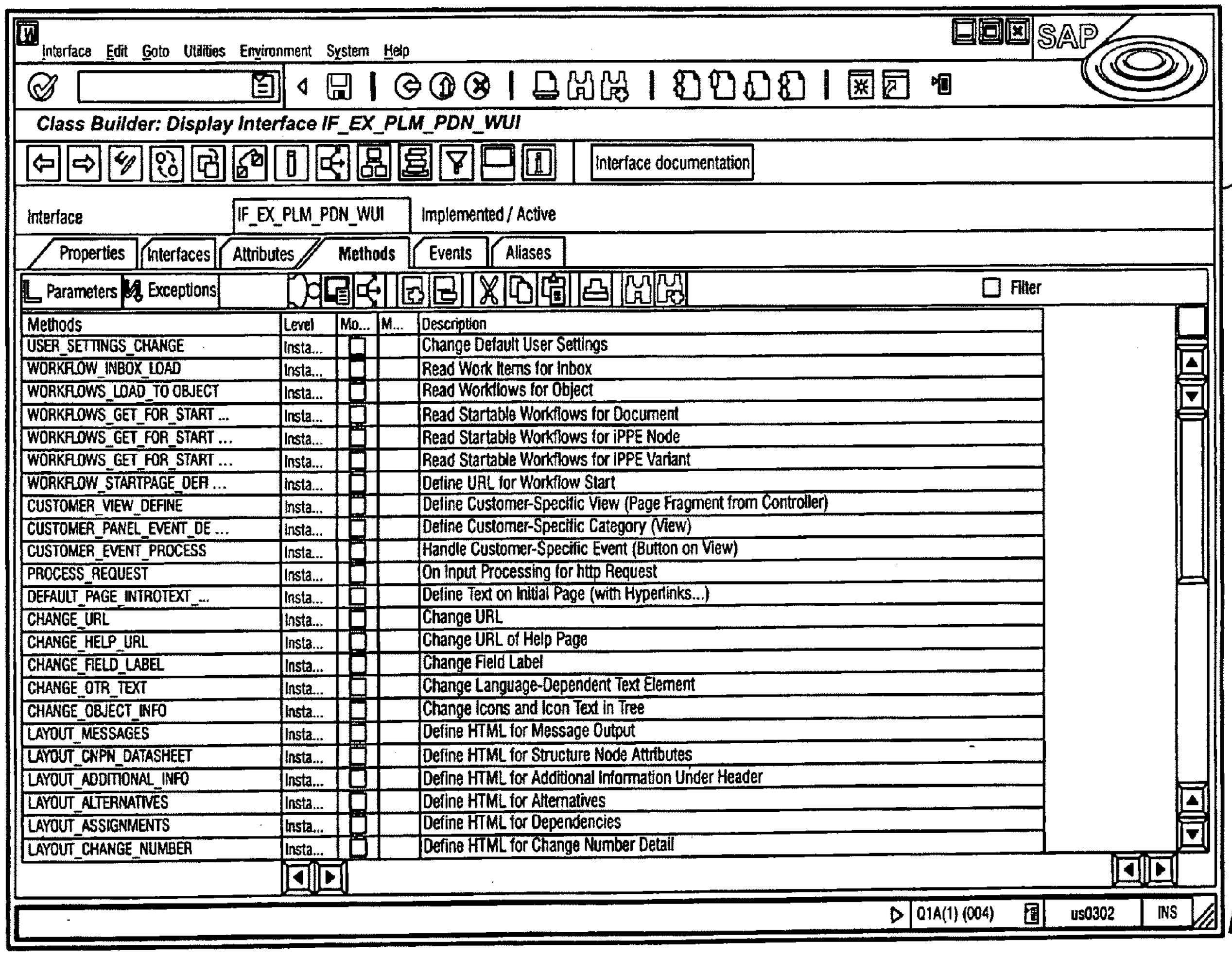

610
Present a User With Changes Available
620
Receive User Instructions to Change a Section of the User Interface Layout
630
Add the User Instructions to Instructions for Displaying the User Interface

FIG. 6

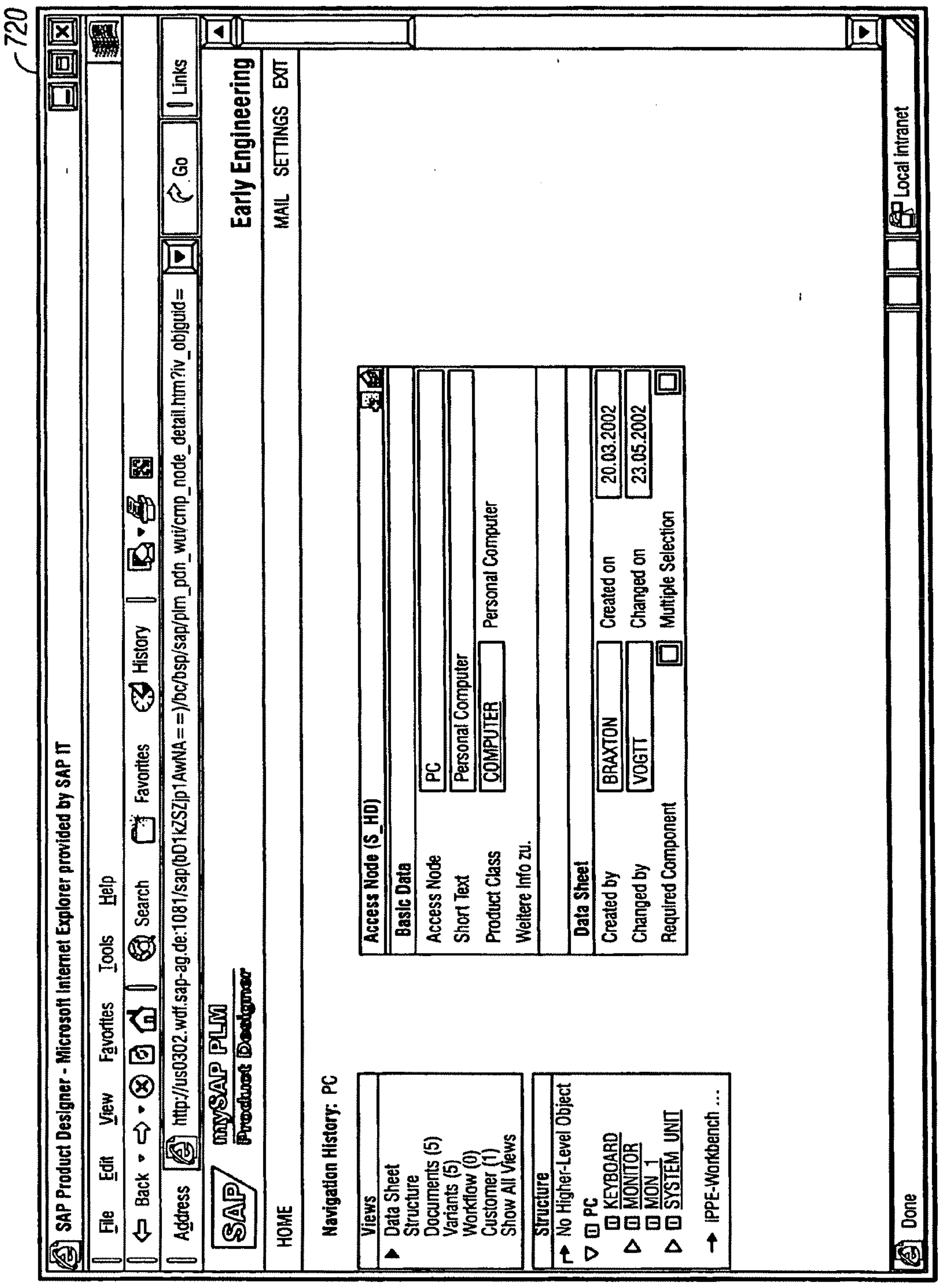


Class Builder: Display Interface IF_EX_PLM_PDN_WUI

Interface IF_EX_PLM_PDN_WUI Implemented / Active

Methods	Level	Mo...	M...	Description
USER_SETTINGS_CHANGE	Insta...	☐		Change Default User Settings
WORKFLOW_INBOX_LOAD	Insta...	☐		Read Work Items for Inbox
WORKFLOWS_LOAD_TO OBJECT	Insta...	☐		Read Workflows for Object
WORKFLOWS_GET_FOR_START ...	Insta...	☐		Read Startable Workflows for Document
WORKFLOWS_GET_FOR_START ...	Insta...	☐		Read Startable Workflows for iPPE Node
WORKFLOWS_GET_FOR_START ...	Insta...	☐		Read Startable Workflows for IPPE Variant
WORKFLOW_STARTPAGE_DEFI ...	Insta...	☐		Define URL for Workflow Start
CUSTOMER_VIEW_DEFINE	Insta...	☐		Define Customer-Specific View (Page Fragment from Controller)
CUSTOMER_PANEL_EVENT_DE ...	Insta...	☐		Define Customer-Specific Category (View)
CUSTOMER_EVENT_PROCESS	Insta...	☐		Handle Customer-Specific Event (Button on View)
PROCESS_REQUEST	Insta...	☐		On Input Processing for http Request
DEFAULT_PAGE_INTROTEXT_...	Insta...	☐		Define Text on Initial Page (with Hyperlinks...)
CHANGE_URL	Insta...	☐		Change URL
CHANGE_HELP_URL	Insta...	☐		Change URL of Help Page
CHANGE_FIELD_LABEL	Insta...	☐		Change Field Label
CHANGE_OTR_TEXT	Insta...	☐		Change Language-Dependent Text Element
CHANGE_OBJECT_INFO	Insta...	☐		Change Icons and Icon Text in Tree
LAYOUT_MESSAGES	Insta...	☐		Define HTML for Message Output
LAYOUT_CNPN_DATASHEET	Insta...	☐		Define HTML for Structure Node Attributes
LAYOUT_ADDITIONAL_INFO	Insta...	☐		Define HTML for Additional Information Under Header
LAYOUT_ALTERNATIVES	Insta...	☐		Define HTML for Alternatives
LAYOUT_ASSIGNMENTS	Insta...	☐		Define HTML for Dependencies
LAYOUT_CHANGE_NUMBER	Insta...	☐		Define HTML for Change Number Detail

FIG. 7A



Access Node (S_HD)

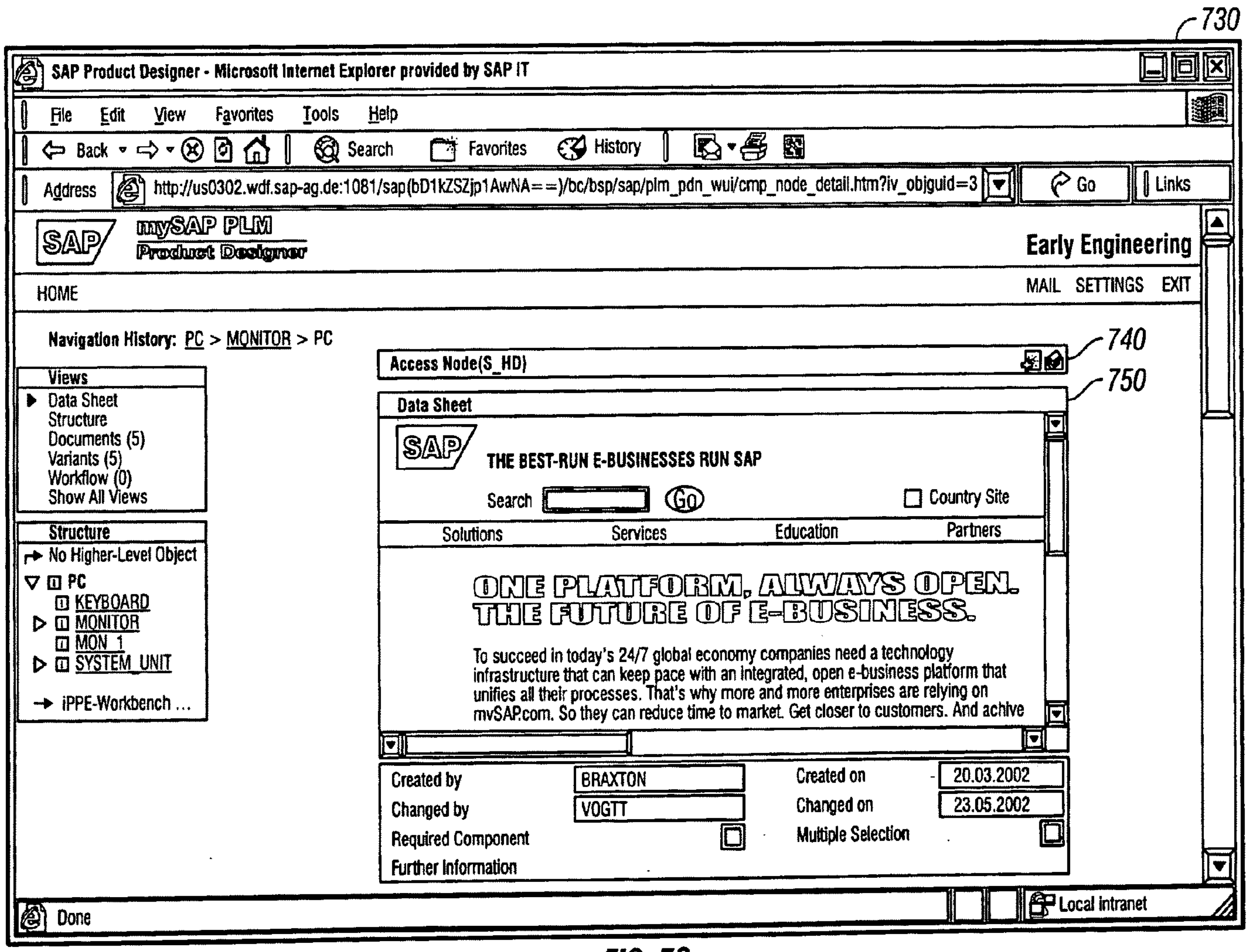
Basic Data

Access Node	PC	
Short Text	Personal Computer	
Product Class	COMPUTER	Personal Computer
Weitere Info zu.		

Data Sheet

Created by	BRAXTON	Created on	20.03.2002
Changed by	VOGTT	Changed on	23.05.2002
Required Component	☐	Multiple Selection	☐

FIG. 7B



http://us0302.wdf.sap-ag.de:1081/sap(bD1kZSZjp1AwNA==)/bc/bsp/sap/plm_pdn_wui/cmp_node_detail.htm?iv_objguid=3

Navigation History: PC > MONITOR > PC

To succeed in today's 24/7 global economy companies need a technology infrastructure that can keep pace with an integrated, open e-business platform that unifies all their processes. That's why more and more enterprises are relying on mySAP.com. So they can reduce time to market. Get closer to customers. And achive

Created by	BRAXTON	Created on	20.03.2002
Changed by	VOGTT	Changed on	23.05.2002
Required Component	☐	Multiple Selection	☐
Further Information			

FIG. 7C

CONFIGURABLE AND EXTENSIBLE USER INTERFACE

BACKGROUND

[0001] The present invention relates to data processing, and more particularly to user interfaces.

[0002] Generally, a data structure is a collection of organized data. Some data structures are represented as hierarchies in which nodes represent data objects and connections between the nodes represent the relationships between the data objects. One type of hierarchy is a tree hierarchy in which there is a root data object from which other data objects branch. Another type of hierarchy is a network hierarchy in which there need not be a root data object. These representations are usually, but need not be, graphical in nature.

[0003] A data structure can include nodes that are business objects, which are data objects that relate to one or more processes of an enterprise. Business objects can represent, for example, business partners, equipment, material master records, products, and so forth.

[0004] A data structure can be implemented as a product structure. A product structure refers to a collection of data objects that relate to or is associated with a particular product. A product structure can be used to model and configure a configurable product such as, for example, an automobile or a computer. As with other data structures, the data objects of a product structure can have complex relationships that can be represented by a hierarchy, such as a tree hierarchy. The data objects of the product structure can include any type of data objects such as, for example, documents, bills of materials, and product attributes.

[0005] Product structures that are included as part of a product design or product modeling application can generally be viewed by a user through a user interface. In certain implementations, the user interface is a Web-based interface established between a client computer and a server computer. A portal is one example of a user interface.

[0006] A data structure can also be implemented as, for example, an organization structure and a project hierarchy. The organization structure can indicate relationships between or among units of an organization. For example, an organization structure of an enterprise can indicate that the engineering department includes a development group and a test group. The project hierarchy can indicate relationships between and among components of a project. For example, an organization hierarchy of a product development project can indicate that a component that represents engineering tasks includes another component that represents testing tasks. Like the product structure, the organization structure and project hierarchy can also be included as part of a computer application.

SUMMARY

[0007] The present invention provides methods and apparatus, including computer program products, for a configurable and extensible user interface.

[0008] In general, in one aspect, a method for changing a user interface layout includes presenting a user with one or more exits points into which user instructions for changes to a user interface can be inserted. The method includes receiving user instructions to change a user interface. The method includes receiving a selection of an exit point. The method includes adding the user instructions to instructions for presenting the user interface at the exit point selected.

[0009] In general, in another aspect, a system enabling user changes to user interface layouts includes a data store storing instructions for displaying a first user interface and instructions for displaying a second user interface including available changes for changing a layout of the first user interface. The system includes a processor for displaying the second user interface to a user, receiving user instructions for changing the first user interface through selections and entries made through the second user interface, and displaying the first user interface to reflect the user instructions received.

[0010] In general in another aspect, a computer-program product for changing a user interface includes instructions operable to cause a processor to identify one or more exits points in a set of instructions that define a user interface. The exit points are locations in the set of instructions into which user instructions for changes to a user interface can be inserted.

[0011] The product includes instructions to present a user with one or more of the exit points identified. The product includes instructions to receive user instructions to change a user interface. The product includes instructions to receive a selection of an exit point. The product includes instructions to add, at the exit point selected, the user instructions to the set of instructions that define the user interface. The product is tangibly stored on machine-readable medium.

[0012] The invention can be implemented to realize one or more of the following advantages. A configurable and extensible user interface allows users to influence the layout of the user interface, which can be, for example, an interface in product design software. Configurability of the interface allows users to insert content desired by the user in some sections while suppressing content of a standard user interface in other sections.

[0013] The user interface can be customized by one or more users, including those who are not developers of the user interface application. For example, the user interface can be customized by an enterprise that is a customer of a provider of the user interface application. As used in this specification, the term user can include an administrator or a developer who were not involved in the development of the user interface application but who are responsible for configuring the user interface for a customer.

[0014] A navigation history allows a user navigating through a product structure to see what objects were previously viewed in a user session. The navigation history also allows the user to return to the object by selecting the object in the navigation history. Selection by, for example, selecting an object of interest from the navigation history results in the display of the selected object. Thus, a user need not back track until the object of interest is displayed and returning to a previously selected object is greatly facilitated. Navigation history also can provide an overview perspective of a product structure and, furthermore, can indicate to the user the position of the object currently selected by the user relative to the rest of the product structure.

[0015] The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. **1** is a block diagram illustrating a system for displaying a user interface.

[0017] FIG. **2** is a block diagram illustrating a product record in a product design application.

[0018] FIG. **3** is a screen shot of a user interface for allowing a user to manipulate a product record.

[0019] FIG. **4** is a flow diagram illustrating a method for presenting a user with a navigation history in a user interface.

[0020] FIGS. **5**A and **513** show screen shots illustrating a user interface having a navigation history.

[0021] FIG. **6** is a flow diagram illustrating a method for changing a layout in a user interface.

[0022] FIGS. **7**A-C show screen shots illustrating controls for changing layout in a user interface.

[0023] Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0024] An object user interface can present users of an application having a plurality of objects with views of the objects. An example of a system having a plurality of objects is a product design application, such as the Product Designer™ application, a component of the R/3™ Enterprise Extension Set™ application distributed by SAP of Walldorf, Germany. The user interface can include a navigation history to allow users to see what objects of the product design process the user previously viewed. The user interface can also be configurable to allow users to change the layout of the user interface.

[0025] As shown in FIG. **1**, an object user interface can be implemented in a product design system **100**. The product design system can include a processor **110** and a data store **120**. The data store **120** can store applications, such as the Product Designer™ application. The data store **120** can also store product records, described below with reference to FIG. **2**, and instructions for displaying user interfaces.

[0026] An example of a product record is shown in FIG. **2**. Data for a product design process can be stored in records in data store **120** (FIG. **1**). The product design data can include objects associated with the product. These objects can include data sheets, product structure, documents, variants, workflow, and so forth. The objects associated with the product product lines, or classes of products. The product record can include a product structure **210** linked to a collection of views such as view **240**. Each view includes links such as, for example, link **212**, to associated objects. For example, the documents view can be linked to documents **250** by link **212**.

[0027] A product structure **210** can be arranged in a variety of ways. In one implementation, the product structure **210** can be organized in a tree structure, as shown. The product associated with the product structure **210** can include a specific product or a product line. The product structure **210** can include one or more components such as, for example, components **220** and **225**. Components can include parts of the product structure or services. Each component can include one or more sub-components such as, for example, sub-components **230** and **235** of the component **225**. Each sub-component can include further sub-components (not shown). For example, if the product **210** is a PC, components can include a keyboard or a monitor. Sub-components of the monitor can include a CRT or a housing.

[0028] The product structure **210** can also include variant structures for each component or sub-component. For example, if a component is a brake for a car, the variant structures can include (1) Disc and (ABS or EDS or SER), (2) Drum and SER, and (3) Drum and (ABS or EDS).

[0029] Each component and/or sub-component of the product structure **210** can be linked to a collection of views such as, for example, view **240**. In the example shown in FIG. **2**, the record for monitor **120** is linked to views **240**. Each view is linked to associated objects, such as documents **250**.

[0030] Links between objects of the product record **200** can include "part of" links or associations. For example, for a PC, a monitor can include a "part of" link to the PC, while a document relevant to the monitor can include an association link to the monitor.

[0031] The product record **200** can be presented to a user through a user interface ("UI") **300**, as shown in FIG. **3**. The user interface can include a structure area **310** listing components of the product structure **210**, a views area **320** listing the available views for the product structure **310**, and a work space **330** displaying the component and view on which the user is working.

[0032] The UI **300** can also include a navigation history area **340**, listing the objects that a user has accessed. In the screen shot **300**, the navigation history displays that the user has accessed the "PC" object record. The navigation history **340** can be presented as a breadcrumb trail, as shown, or as other structure types. The navigation history **340** is updated each time a user accesses a new object. Each object listing in the navigation history **340** contains a link to the object, allowing a user to return to the object by selecting the object listing. The navigation history can be stored in persistent memory as a table.

[0033] FIG. **4** is a flow diagram illustrating one implementation of a method for presenting a user with a navigation history in UI **300**. System **100** receives a user selection of an item in the product record **200** (step **410**). The user can submit a selection to system **100** by selecting an item presented through UI **300**. For example, the user can select "Monitor" from structure area **310**. UI **300** can include a linked subset of objects in the product record. The objects displayed in a given screen can include objects or controls to navigate to objects that are part of or associated with an object of the product record displayed. For example, the screen shot of FIG. **3** displays objects (e.g., "Keyboard," "Monitor") that are part of a PC record, and ways to navigate (e.g., "Views" **320**) to other objects that are associated with the PC record. The page representing the object at the highest level of a product structure can be referred to as the home page of the object. The home page of the object can be accessed by a user through a link sent to the user or as a result of a search done by the user.

[0034] System **100** determines if the user selection contains an object identification, i.e., "object id" (step **420**). An object identification indicates that the user selected an object to view. If the user selection does not contain an object id, the system returns to step **410** to await another user selection.

[0035] If the user selection does contain an object id, system **100** determines if the object id matches the object id of the last selected object (step **430**). If the object id is the same as the object id of the last selected object, system **100** returns to step **410** to await another user selection. If the object id does not match the object id of the last object selected, system **100** updates the navigational history area **340** in user interface **300** to reflect the new object id. (step **440**). The object id is updated independently of the Web page address, if the UI **300** is part of a Web page having an address. Thus, if the Web page address does not change while the object currently selected changes, navigation history **340** will be updated to reflect the change in object. Conversely, if the Web page address changes while the object being viewed does not change, navigation history **340** will not be updated.

[0036] FIGS. **5**A and **58** are screen shots illustrating a user interface having a navigation history **340**. Screen shot **510** illustrates a navigation history that indicates that the user has selected the objects "PC," "Monitor," and "FD_DMU." Although the user also selected a documents view before selecting document "FD_DMU" **150**, the documents view does not show in the navigation history.

[0037] Screen shot **520** illustrates a navigation history in which the user returned to the "PC" screen before selecting "Keyboard." System **100** includes "PC" twice in navigation history **340** since the user selected the "PC" screen again after viewing "FD_DMU." Since there was a change in object in between the two viewings of the "PC" screen, system **100** records the path the user took.

[0038] System **100** can update the navigation history **340** by storing the object id of the object with the object type of the object. The object type and object id of the object can be stored in a table. The table can be stored in a memory, if the application including user interface **300** is a stateful application (i.e., having memory). The table can be stored in a server side cookie if the application including user interface **300** is a stateless application (i.e., having no memory). The server side cookies can be saved from one response/request cycle to the next. The table can be deleted from storage at the end of a user session.

[0039] FIG. **6** is a flow diagram illustrating one implementation of a method for changing layout in a user interface. System **100** presents a user with changes available for a UI **300** (step **610**). A user can be presented with the available changes through a second UI, such as the UI **710** shown in the screen shot of FIG. **7**A. As illustrated in UI **710**, a user can change URLs, field labels, text elements object information, and other layout information. The methods that start with "LAYOUT" can be used to add hypertext markup language ("HTML") code or to suppress the standard layout of the UI **300**.

[0040] System **100** can receive user instructions to change a section of the UI **300** layout (step **620**). Receiving instructions can include user selection of controls for pre-defined changes or receiving code, such as HTML code. As shown in screen shot **720** of FIG. **7**B, a customer view labeled "Customer **1**" has been added to the view selector. Also, the label of the field next including the entry "Personal Computer" has been changed to "Short text." This can be done through coding such as the example shown in Table 1.

TABLE 1

```
method IF__EX__PLM__PDN__WUI~CHANGE__FIELD__LABEL.
  if iv__dataelement = ‘PVS__PNODETX’.
    cv__label = ‘Short text’.
  endif.
endmethod.
```

[0041] System **100** can add the received instructions to the existing instructions for presenting UI **300** to a user (step **630**).

[0042] Another example of changed layout is shown in screen shot **730** of FIG. **7**C. For this change, coding in a first group **740** box of area **330** has been suppressed using coding such as the example shown in Table 2.

TABLE 2

```
method IF__EX__PLM__PDN__WUI~LAYOUT__OBJECT__HEADER.
  ev__suppress__group = ‘X’.
endmethod.
```

[0043] In a second group box **750**, HTML has been added to insert an iframe, showing another web page, at the beginning of the group. The iframe and a link labeled "Further Information" is added using coding such as the example shown in Table 3.

TABLE 3

```
concatenate ‘<table><tr><td>’
        ‘<iframe src=“http://www:sap.com” width=“500” ’
        ‘height=“300” frameborder=“0” ’
        ‘align=“left” valign=“top” scrolling=“yes” >’
        ‘</iframe>’
        ‘</td></tr></table>’
  into ev__html__string__top.
ev__html__string__bottom = ‘<a href=“#1”>Further Information...</a>’.
```

[0044] The instructions can also include changing icons or text associated with an icon. The instructions can be inserted at any exit point of an application. The instructions can be inserted at the beginning of a page fragment or the end of a page fragment.

[0045] The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The invention can be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device or in a propagated signal, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

[0046] Method steps of the invention can be performed by one or more programmable processors executing a computer program to perform functions of the invention by operating on input data and generating output. Method steps can also be performed by, and apparatus of the invention can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit).

[0047] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

[0048] To provide for interaction with a user, the invention can be implemented on a computer having a display device such as a CRT (cathode ray tube) or LCD (liquid crystal display) monitor for displaying information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, such as visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0049] The invention can be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or an Web browser through which a user can interact with an implementation of the invention, or any combination of such back-end, middleware, or front-end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN"), a wide area network ("WAN"), and the Internet.

[0050] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0051] The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the steps of the invention can be performed in a different order and still achieve desirable results. The to systems and methods described can be applied to any user interface that allows modification, either statically or dynamically, at run time. The systems and methods can be applied to, for example, Web-based user interfaces. The Web-based interface can use page fragments and any markup language such as, for example, JavaServer Pages™, Active Server Pages, and Business Server Pages. Alternatively, the user interface described can be implemented by using servlets or any HTTP server, which content can include instructions derived from programming languages such as, for example, C++, COBOL, JAVA, and Advance Business Application Programming ("ABAP"). Any part of the user interface can be changed, including, for example, field labels, text, hyperlinks, and Multipurpose Internet Mail Extensions ("MIME") objects. A user, as used in this specification, refers to any combination of a human operator or a computer-program product. The user can be an administrator, developer, or programmer who may not have been involved with developing the user interface application but who are responsible for customizing the user interface for an enterprise that is a customer of the provider of the user interface application. The methods and apparatus described are not limited to product structures but, rather, can be applied to any structures of business objects, such as, for example, an organization structure or a project hierarchy.

What is claimed is:

1-19. (canceled)

20. A method, comprising:

presenting a first user interface for displaying views of one or more objects in a product structure associated with a tangible product;

presenting a second user interface comprising controls for implementing pre-defined changes to the first user interface and a console for accepting a user-defined coding for implementing user-defined changes to the first user interface;

receiving at the console, a textual user-defined coding including lines of code defined by and entered by the user, the lines of code comprising instructions to implement a change to at least a portion of an existing layout of the first user interface;

providing, through the second user interface, an ability to, in response to the textual user-defined coding, suppress one or more sections of a standard layout of the user interface; and

adding the textual user-defined coding to textual instructions for presenting the first user interface, the textual user-defined coding being added at a user-selected location in the textual instructions of existing computer code, wherein adding modifies the existing layout of the user interface in accordance with the instructions for implementing the change.

* * * * *