

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 September 2001 (20.09.2001)

PCT

(10) International Publication Number
WO 01/69380 A2

- (51) International Patent Classification⁷: G06F 9/44
- (21) International Application Number: PCT/US01/08151
- (22) International Filing Date: 14 March 2001 (14.03.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:

60/189,191	14 March 2000 (14.03.2000)	US
09/610,179	5 July 2000 (05.07.2000)	US
09/610,181	5 July 2000 (05.07.2000)	US
09/610,186	5 July 2000 (05.07.2000)	US

(81) Designated States (*national*): AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

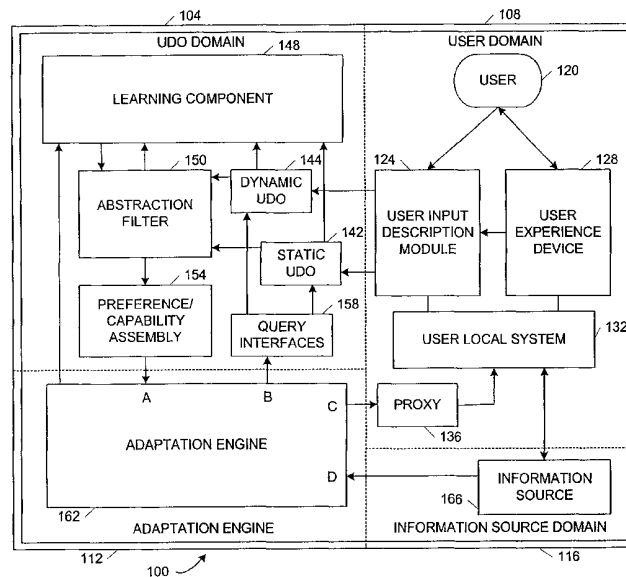
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

- (71) Applicant: EDAPTA, INC. [US/US]; 11343 Trailside Way, San Diego, CA 92121 (US).
- (72) Inventors: MCCOY, Wayne; 20015 Haller Avenue, Poolesville, MD 20837 (US). WILDER-SMITH, Chris; 69 Rockland Street, Natick, MA 01760 (US).
- (74) Agent: ISRAESEN, Ned, A.; Knobbe, Martens, Olson & Bear, LLP, 16th Floor, 620 Newport Center Drive, Newport Beach, CA 92660 (US).

Published: — without international search report and to be republished upon receipt of that report

[Continued on next page]

(54) Title: A SYSTEM AND METHOD FOR ENABLING DYNAMICALLY ADAPTABLE USER INTERFACES FOR ELECTRONIC DEVICES



(57) Abstract: Information systems, including computers, network appliances, home appliances, personal digital assistants, set-top boxes and smartphones are enhanced to include software and communications processes that permit conveying user preference and capability information to an adaptation service for the purpose of generating and returning transformations of user interfaces and content of applications and information necessary for operational configuration and setup. The system provides a universal enabling of user interfaces and content that can adapt to user capabilities, preferences and situations. In particular, this invention enables the information system environment for every individual user, regardless of situation or capability.

WO 01/69380 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

**A SYSTEM AND METHOD FOR ENABLING DYNAMICALLY
ADAPTABLE USER INTERFACES FOR ELECTRONIC DEVICES**

Background of the Invention

5 Field of the Invention

The field of the invention relates to information technology systems and devices. More particularly, the field of the invention relates to dynamic adaptation of user interfaces and information content associated with such equipment.

Description of the Related Technology

10 The march of technology makes possible appliances that yesterday were only science-fiction dreams. The development of assistive technology is a catalyst for discovery of new ways to make computing easier for everyone. For example, voice recognition is useful for a worker with busy hands, and it is crucial for someone without any hands. Eye-tracking pointing devices are useful for many applications and mandatory for some physically challenged individuals. Thus, the size of the market for any assistive technology is not confined to those with disabilities: It
15 includes others that have special needs due to new applications of computing in the workplace and the home.

Assistive technology can be used to help a person who is hindered in some way from interacting with a system in conventional ways. However, these assistive technology solutions are not very extensible and do not adapt to the changing needs of a user.

For example, assistive technology has developed voice synthesis and recognition programs to help individuals
20 that have a disability with their hands interact with computing devices. However, known systems do not adjust the output level of the synthesized voice based upon the level of the background noise. Thus, in conditions of significant noise, users of the system may have difficulty hearing the output. Furthermore, in conditions of low noise, the voice synthesis may be too loud for comfort.

Thus, individuals are in need of a system that adapts the interfaces to electronic devices based upon
25 detected operating context conditions. The system should allow the user to specify their preferences with respect to the interfaces. Furthermore, the adaptation of the interface should be automatically performed in response to the user accessing the electronic device.

Summary of the Invention

One aspect of the invention includes a method of adapting a predefined interface, the method comprising
30 identifying a preference object that includes at least one user characteristic from a set of user characteristics, wherein the set of user characteristics comprises at least one of situational information, environmental information, behavior information, and context information, transmitting a first capability object from a selected electronic device to an adaptation engine, wherein the first capability object defines one or more attributes regarding the electronic device, transmitting a second capability object to the adaptation engine, wherein the second capability object defines one or
35 more attributes of an information source that is accessed by the electronic device, and generating an adaptation

object, based at least in part upon the at least one user characteristic, the contents of the transmitted first capability object, and the contents of the transmitted second capability object, wherein the adaptation object defines at least one adaptation rule for adapting human factor information that is transmitted by the electronic device or the information source.

5 Another aspect of the invention includes an adaptation engine, comprising a communication interface configured to receive a plurality of preference objects and a plurality of capability objects, wherein the preference objects define at least one preference of a user, and wherein the capability objects each define the capabilities of a plurality of electronic devices, and at least one adapter service unit that generates, based at least in part upon the contents of the preference objects and the contents of the capability objects, a set of rules for modifying one or more
10 aspects of an interface between the electronic devices and the user.

 Another aspect of the invention includes an adaptation object residing in a memory of an electronic device, the adaptation object comprising a plurality of adaptation instructions including at least one rule that specifies a change to an interface between a user and an electronic device, wherein the adaptation object is generated in real time based in response to the user requesting to use a computing device to retrieve information from an information source,
15 and wherein the contents of the adaptation object are based at least in part upon the content of a capability object that is associated with the electronic device, the content of a capability object that is associated with the information source, and the content of a preference object that defines one or more preferences of the user.

 Another aspect of the invention includes a system for adapting a predefined interface, the system comprising means for selecting at least one user characteristic from a set of user characteristics, wherein the set of user
20 characteristics comprises situational information, environmental information, behavior information, and context information, means for transmitting a first capability object from a selected electronic device to an adaptation engine, wherein the first capability object defines one or more attributes regarding the electronic device, means for transmitting a second capability object to the adaptation engine, wherein the second capability object defines one or more attributes of an information source that is accessed by the electronic device, and means for generating, based at
25 least in part upon the at least user characteristic, the transmitted first capability object, and the transmitted second capability object, an adaptation object, wherein the adaptation object defines at least one adaptation rule for adapting human factor information that is transmitted by the electronic device or the information source.

 Another aspect of the invention includes a method of generating an adaptation object, the method comprising identifying one or more user preferences, identifying the capabilities of a first device, identifying the capabilities of a
30 second device, and determining whether to modify an interface of the first device, wherein the determining is based at least in part upon the identified user preferences, the identified capabilities of the first device, and the identified capabilities of the second device.

 Another aspect of the invention includes an adaptation system comprising a user input description module for receiving preference information that defines preferentially one or more types of presentation that are preferred by the
35 user, a preference assembly module for building a preference object that is based at least in part upon the preference

information, an electronic device that is accessible by the user, an electronic device capability object that is associated with the electronic device, wherein the electronic device capability objects defines the capabilities that are associated with the electronic device, an information source that is accessible by the electronic device, an information source capability object that is associated with the information source, wherein the information source capability object
5 defines the capabilities that are associated with the information source, and an adaptation engine for generating, based at least in part upon the preference object, the electronic device capability object, and the information source capability object, an adaptation object, wherein the adaptation object defines at least one adaptation rule for adapting human factor information that is transmitted by the electronic device or the information source.

Another aspect of the invention includes a program storage device storing instructions that when executed
10 perform the acts comprising identifying one or more user preferences, identifying the capabilities of a first device, identifying the capabilities of a second device, and determining whether to modify an interface of the first device, wherein the determining is based at least in part upon the identified user preferences, the identified capabilities of the first device, and the identified capabilities of the second device.

Another aspect of the invention includes a preference object residing in a memory of an electronic device, the
15 preference object comprising at least two preference functions, wherein each of the at least two preference functions have an associated type, wherein the type is one of: entry, control, presentation, and authorization, at least one preference rating for each of the preference functions, wherein the preference rating preferentially orders the at least two preference functions.

Another aspect of the invention includes a method of generating a preference object in an electronic device,
20 the method comprising receiving a user description object that defines user characteristic information, wherein the user characteristic information is selected from the group comprising: situational information, environmental information, behavior information, and context information, storing the user description object in a database, determining the operating contextual conditions of the user, and generating, in response to a user accessing an electronic device, a preference object that defines one or more user preferences, wherein the content of the preference
25 object is at least based in part upon the content of the user description object and the determined operating contextual conditions.

Another aspect of the invention includes a program storage device storing instructions that when executed performs the acts comprising receiving a user description object that defines user characteristic information, wherein the user characteristic information is selected from the group comprising: situational information, environmental
30 information, behavior information, and context information, storing the user description in a database, determining the operating contextual conditions of the user, and generating, in response to a user accessing an electronic device, a preference object that defines one or more user preferences, wherein the content of the preference object is at least based in part upon the content of the user description object and the determined operating contextual conditions.

Another aspect of the invention includes a system for generating a preference object in an electronic device,
35 the method comprising means for receiving a user description object that defines user characteristic information,

wherein the user characteristic information is selected from the group comprising: situational information, environmental information, behavior information, and context information, means for storing the user description in a database, means for determining the conditions of the user, and means for generating, in response to a user accessing an electronic device, a preference object that defines one or more user preferences, wherein the content of the preference object is at least based in part upon the content of the user description object and the determined conditions.

Yet another aspect of the invention includes a capability object residing in a memory of an electronic device, the capability object comprising a descriptor which identifies an electronic device or software application that is associated with the capability object and which identifies whether the electronic device or software application is directly accessible by the user or, alternatively, an information source that is accessible by the electronic device or software application, and at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object as being either (1) an electronic device or software application or (2) an information source.

Another aspect of the invention includes a method of making a capability object, the method comprising storing, in a memory, a descriptor which identifies an electronic device or a software application that is associated with the capability object and which identifies whether the electronic device or software application is associated is accessible by the user or, alternatively, an information source that is accessible by the electronic device or software application, and storing, in the memory, at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object.

Another aspect of the invention includes a system for making a capability object, the system comprising means for storing, in a memory, a descriptor which identifies an electronic device or a software application that is associated with the capability object and which identifies whether the electronic device or software application is accessible by the user, or, alternatively, an information source that is accessible by the electronic device, and means for storing, in the memory, at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object.

Brief Description of the Drawings

Figure 1 is a block diagram illustration an adaptation system.

Figure 2 is a block diagram illustrating the components of an adaptation engine that is shown in Figure 1.

Figure 3 is a high level flowchart illustrating an adaptation process of the present invention.

Figures 4A, 4B, 4C, and 4D are collectively a flowchart illustrating in further detail the adaptation process of Figure 3.

Figures 5A and 5B are collectively a flowchart illustrating a process for generating an adaptation object using the adaptation engine of Figure 2.

Detailed Description of Embodiments of the Invention

The present invention has several aspects, no single one of which is solely responsible for its desirable attributes. Without limiting the scope of this invention as expressed by the claims which follow, its more prominent features will now be discussed briefly.

5 Figure 1 is a block diagram illustrating certain components of an adaptation system 100. The adaptation system 100 comprises four functional domains: (i) a user domain 104, (ii) a user description object (UDO) domain 108, (iii) an adaptation domain 112, and (4) an information source domain 116. A summary of the components of each of the domains is set forth immediately below. A more detailed description of each of the components is described further below in Section I, entitled "System Components".

10 The user domain 104 includes the immediate computerized environment that is associated with a user 120. The user domain 108 comprises: a user 120, a user input description module 124, a user experience device 128, a user local system 132, and a user proxy 136.

 The UDO domain 108 includes a set of devices that are used to receive information about the user, the user preferences, the electronic devices that are being accessed by the user. The UDO domain is used to generate a preference object that describes the user's 120 preferences. Furthermore, the UDO domain 108 is used to generate or retrieve a capability object that is associated with the user local system 132. The UDO domain comprises a static UDO 142, a dynamic UDO 144, a learning component 148, an abstraction filter 150, and a preference/capability assembly 154. The adaptation domain 112 includes an adaptation engine 162 that is used to customize the interface between the user 120 and the user local system 132. The information source domain 116 includes an information source 166. The information source contains data or services that may be requested by the user 120 via the user local system 132.

 The domains 104, 108, 112, and 116 may each reside on a single electronic device, or alternatively, it is possible for all or selected ones or all of the domains to reside on separate electronic device. The term electronic device can include any device that is capable of receiving or transmitting information. The electronic device may include: a computer; a personal appliance; an ATM; a kiosk; a handheld device; a smart appliance, a network of devices, a networked federation of computers; a game system; electronic instrumentation; an automobile; a television; a telephone; a lamp; an air conditioning system; a sprinkler system; an elevator; or a monitoring and control system for a room, such as family room, an office, or an elevator.

 The electronic device may comprise any conventional general purpose single or multi-chip microprocessor, such as (but not limited to) Intel Pentium-class X86 processors and clones, AMD Athlon, Motorola 68K and Power PC, Sun SPARC, C-Cube microSPARC, Hitachi SuperH, Systems-on-a-chip including Lucent ARM products and Intel StrongARM, and Motorola M-CORE products, or any improvements or extensions to these, bearing an operating system and environment, such as (but not limited to) Windows 95, Windows 98, Windows NT/2000, Windows ME, Windows CE, Linux, UNIX and UNIX variants (BSD, SCO, AT&T, AIX, HP-UX, and the like), Solaris, PalmOS, Microware OS-9, and Apple MacOS, Sun (and others) Java or any improvements or extensions to these. In addition,

the electronic device may be any special purpose or embedded microprocessor, such as a digital signal processor, or any processor with the operating/control environment provided in firmware or hardware.

One embodiment of the adaptation system 100 uses three different kinds of objects: (1) preference objects, (2) capability objects, and (3) adaptation objects. The preference object identifies the user and provides a set of usage preferences for the user. The preference object contains rules about the preferences and are interpreted based on the current time and situation by the adaptation engine 162. The capability object identifies the capabilities of the user local system 132 and the information source 166. The adaptation object contains a set of rules that are used to transform the interface between the user 120 and the information source 166 such that the user is presented with an optimum user-computer interaction for the current situation. Preference and capability objects can be mobile objects and may be generated in various languages depending on the needs of the end-user environment. The structures of preference object, capability object and adaptation objects are discussed farther below in the section II, entitled, "Formal Object Syntax."

For convenience of description, the following text refers to two types of capability objects: a user capability object and an information source capability object. The term "user capability object" is used below to refer to the capability object that is associated with the user local system 132. The term "information source capability object" is used below to refer to the capability object that is associated with the information source 166.

Set forth below is a detailed description of each of the components of the adaptation system 100. The following description is divided in the following sections: I. System Components, II. Formal Object Syntax, III. Method of Operation, and IV. Illustrative Examples.

I. System Components

The description of the system components is divided into the following sub-sections: 1. The User, 2. The User Input Description ; 3. The User Experience, 4. The User Local System, 5. The Static UDO, 6. The Adaptation Engine, 7. The Query Interface, 8. The Dynamic UDO, 9. The Abstraction Filter, and 10. Preference Object/Capability Object Assembly, 11. User's Proxy, 12. Learning Component, and 13. Information Source.

1. The User

By way of the user experience device 128, the user 120 receives presentation material from the user local system 132 and provides control and input actions to the user local system 132. The user input description module 124 can detect the operating context conditions of the user 120. Conditions that may be detected include:

- Situational – what the user may be involved in doing: walking, sitting, riding or driving, carrying things, hands in messy material, task;
- Environmental – weather conditions, lighting, noise, time of day, interfering activities; computing environment (standalone, hand-held, networked, operating system, application, etc.)
- Behavioral – fatigue, alertness, stress, physical strain, discomfort; and
- Contextual – where the user is: home, in a car, office, on a street, in church.

35

2. The User Input Description

The user input description module 124 is in communication with sensors and other input devices for gathering operating context data about the user 120. Furthermore, the user input description module 124 gathers preference information from the user 120. The user input description module 124 may be any software or hardware
5 designed to capture user description input from the user, for example, directly on a personal computer, input to a website, or in the form of a questionnaire. Such information may be gathered while the user is directly interacting with the adaptation system 162 or it may be have been previously gathered in some prior activity. As an example of the latter, a service organization, set up explicitly for the purpose, could gather the information in support of the user's purchase of an instance of the user local system 132. The tools also include software or hardware devices that can
10 assess the nature of a person's disability; for example, a computer-based tool that runs simple tests to determine visual acuity, colorblindness, field of view, and the like, for every user not already determined to be blind.

The user input description module 124 is also capable of stimulating the generation of preference objects and capability objects through direct interaction with the static UDO 142 and the dynamic UDO 144 whenever changes in conditions of the user 120, the user experience device 120, or the user local system 132 are significant enough to
15 warrant an adaptation. The adaptation engine 162 is an active collaborator in the session between the user and the information source 166. The preference objects and capability objects are cached in the adaptation engine 162 in a preference object cache 212 and a capability object cache 216 (shown in Figure 2).

Depending on the context, the user descriptions are defined to include as much about a person as is necessary to accommodate the various usage condition of that context. As is discussed further below in Section III,
20 the user description is used to generate a preference object that is associated with the user.

The user description contains information about an individual that qualifies in some manner the use of the user local system 132. For example, persons who are unable to use their legs, but who have use of their hands, may be able to sit at a desk and a desktop computer with no limitations to their ability whatever. However, if they happen to walk using crutches, their ability to use certain handheld devices may be compromised. Thus, if such a person
25 wished to have accommodation for handheld devices, then they would put information about their disability into the user description. If they believe that they will never use a handheld device, for whatever reason, then they could elect to not include any information about their disability. The information will be used to select or compute preferences based upon need, personal likes, situation, environment, context, etc., as discussed earlier.

3. The User Experience

The user experience device 128 presents to the user 120 information that is received by the user local system 132 from the information source 166. Furthermore, the user experience device 128 receives control information from the user 120 regarding services or information that are provided by the information source 166. The user experience device 128 can render and receive visual, aural, tactile, haptic, gestural and other modes, and specifically includes assistive and accessor devices that accommodate a disability of a user. Typically, for a personal
35 computer, the user experience device 128 can include a keyboard, a mouse, a monitor and speakers as devices, and a

graphical user interface (GUI) for the operating system and applications. For a blind user, the user experience device 128 could include a screen reader. The user experience device 128 also includes: ATMs, kiosks, smartphones, set-top boxes, smart appliances, smartcards, Java™ rings, and RF tags.

4. The User Local System

5 The user local system 132 includes an electronic device that the human user may use to retrieve information or requests services from the information source 166. The user local system 132 may also include an operating system, applications, and peripheral devices including assistive and accessor devices. The user local system 132 can include any electronic device. Depending on the embodiment, the user local system 132 may be integrated with the user experience device 128.

10 5. The Static UDO

The static UDO 142 is a database containing elements which describe a user in terms of capabilities and preferences. The static UDO comprises a set of rules relating these data elements to dynamic conditions that qualify usage for the user and to other such rules, and hints that may be employed in making an adaptation for the user. The static UDO 142 includes those elements which either change very slowly over time or not at all. The dynamic UDO 15 144 includes those conditions, capabilities and possibly preferences that can be expected to change over relatively short periods of time. The static and dynamic UDOs are combined and abstracted by the abstraction filter 150 to form a correlation of preference object, which describes to the adaptation engine 162 what the user needs to obtain an optimal interface for using the information source 166. The static UDO 142 includes the following information:

- (i) a unique identifier;
- 20 (ii) abilities and fixed constraints, e.g., disabilities; and
- (iii) fixed system preferences e.g., control, input, and presentation.

UDO objects are generated based upon the contents of the user description that is received from the user input description module 124.

The structure of one embodiment of a user description object is described below using an XML-like example.

25

```
< FACULTY TYPE="mobility" >
  < FUNCTION TYPE="ground" STATUS="assisted" > wheelchair </FUNCTION >
  < FUNCTION TYPE="stair" STATUS="none" > </FUNCTION >
</FACULTY >
```

30

```
< FACULTY TYPE="motor" >
  < FUNCTION TYPE="hand" STATUS="none" > bilateral </FUNCTION >
  < FUNCTION TYPE="leg" STATUS="none" > bilateral </FUNCTION >
</FACULTY >
```

35

```

< FACULTY TYPE="touch" >
    < FUNCTION TYPE="hand" STATUS="none" > bilateral < /FUNCTION >
< /FACULTY >

```

5 In the foregoing example, the user description specifies that the user 120 has a disability. The disability could be caused by cervical injury resulting in loss of effective use of arms and legs and certain body functions below the chest. Note that the disability is characterized in two different way in the user description: mobility and motor. Leg dysfunction usually results in an inability to walk, hence a mobility difficulty. On the other hand, the same injury causes an inability to use the hands effectively, due primarily to the loss of fine motor movements. Likewise, touch
10 senses on the hands are diminished. These all have very different implications for the use of computing resources and devices.

Additional exemplary user description is as follows.

```

< PREF TYPE="entry" LOCUS="accessor" PRIORITY="0" > voice < /PREF >
15 < PREF TYPE="control" LOCUS="accessor" PRIORITY="0" > voice < /PREF >
< PREF TYPE="authorize" LOCUS="accessor" PRIORITY="0" > voice < /PREF >
< PREF TYPE="present" LOCUS="accessor" PRIORITY="0" >
    < VISUAL DEVICE=[Acme reference] UI=[GUI reference] / >
< /PREF >
20 < PREF TYPE="present" LOCUS="accessor" PRIORITY="1" >
    < AURAL DEVICE=[Acme reference] API=[sound API reference] / >
< /PREF >

< PREF TYPE="entry" LOCUS="extra" PRIORITY="0" > voice < /PREF >
25 < PREF TYPE="control" LOCUS="extra" PRIORITY="0" > voice < /PREF >
< PREF TYPE="authorize" LOCUS="extra" PRIORITY="0" > voice < /PREF >
< PREF TYPE="present" LOCUS="extra" PRIORITY="0" > default < /PREF >

```

30 The user's preferences, on the other hand, can be entirely independent of any qualifying conditions that are specified. For example, the preferences are for the same user described above, but they could as easily be for any user with no disability who chooses to operate their "accessor" by voice control and entry, but who views the visual presentation on the accessor's screen. A lower priority choice states that aural presentation in lieu of visual presentation is acceptable. This might be chosen when the person is engaged in some activity where they cannot see the screen, or should not be looking at it, such as while driving a car.

This user has also expressed a preference for using the accessor as a control and entry device for systems external to the accessor ("extra"), while using that system's visual display, if it has one.

6. Adaptation Engine

5 Figure 2 is a block diagram illustrating certain sub-components of the adaptation engine 162. The adaptation engine 162 includes the following modules: a communications interface 204, a capabilities registry 208, a preference object cache 212, a capability object cache 216, an adaptation event manager 220, an adaptation manager 224, a plurality of adapter service units 228, an interpretive consolidator 232, an adaptation service registry 236, an adaptation object assembler 240, a learning engine 244, a session manager 248, and an accounting log 252.

10 The communication interface 204 includes protocols and services for receiving preference objects and capability objects and for transmitting adaptation objects to appropriate recipients. The capabilities registry 208 provides services to transmit capability objects and preference objects.

The preference object cache 212 provides a database of currently active preference objects. The capability object cache 216 provides a database of currently active capability objects. The adaptation event manager 220 receives events from the capabilities registry and initiates the process of generating an adaptation object. The adaptation manager 224 performs first-level negotiation among the associated preference objects and capability objects, selects one or more appropriate adapter service units 228, and passes it the appropriate capability object and preference objects. The selection of an appropriate adapter service unit 228 to be considered may actually be determined by the abstraction filter 150 and indicated in a capability object sent to the adaptation manager to accomplish the transformation for a given type of capability.

20 The adapter service units 228 perform the interpretation of data from capability objects and preference objects to produce the data required to generate an adaptation object. The adapter service units 228 create an adaptation object skeleton (or framework), to be employed by an interpretive consolidator 232 to construct an adaptation protocol. The adaptation protocol object embodies a formal set of rules that describes the transformation. In one embodiment, the adaptation protocol comprises a sequence of adaptation concepts .

25 An adaptation concept is a formal structure of what the adaptation object will represent in a concrete way. An adaptation concept has associated with it, a set of adaptation constraint objects. The adaptation constraint objects give the bounds of the solution, including for example, minimum, maximum and preferred values. In one embodiment, both the adaptation concepts and their adaptation constraints have associated weights that are used in fuzzy logic/neural net approaches that are employed in the interpretive consolidator 232. The format of one embodiment of the adaptation concepts are described in further detail below in Section II.

30 The interpretive consolidator 232 receives the adaptation object skeleton from the adapter service units 228 and applies a set of internal rules and the suggestions of the learning engine 244 to make a broader determination of the required transformations and to eliminate conflicts from the adaptation concepts.

The adaptation service registry 236 comprises information about third-party adaptation services that have registered with the adaptation engine 162. Third party services can be called by the adaptation object assembler to assist in the process of generating the adapting object.

5 The adaptation object assembler 240 uses the adaptation object concepts to generate a complete adaptation object that is sent to the appropriate recipient. The adaptation object assembler 240 interprets the adaptation protocol object and generates an adaptation object that can be sent back to the user local system 132 or to the user's proxy 136. The adaptation object assembler 240 has access to the adaptation service registry 236. In situations where adaptation services of a remote or third party are needed, the adaptation object assembler 240
10 gathers the information required to properly configure the adaptation object to use the third party service from the registry. The adaptation object assembler 240 is also responsible for accounting and logging the events regarding the generation of the adaptation objects.

The learning engine 244 makes inferences on adaptation transactions to optimize adaptation objects. The session manager 244 maintains bookkeeping information about currently active sessions. Session objects correlate
15 the relationship between capability objects, preference objects, and adaptation objects. One embodiment of a session objects is described below in Section II.

The accounting log 252 maintains usage statistics and can be used for transaction accounting, billing, etc.

7. Query Interfaces

The query interfaces 158 provide access to the static UDO 6 and the dynamic UDO 9 for externally
20 stimulating preference object and capability object generation and for storing user descriptions. In one embodiment, for security purposes, the adaptation engine 162 cannot read the UDO information directly. Upon request from the adaptation engine 162, user description information is transmitted from the static UDO and the dynamic UDO to the abstraction filter 150 which creates explicit control, input and presentation statements representing the user description information.

25 8. Dynamic UDO

The dynamic UDO 144 comprises the user description that can be expected to change over relatively short periods of time. The dynamic UDO includes a snapshot of the current conditions surrounding the user 120, the local user local system 132, and the user experience device 128.

9. Abstraction Filter

30 The abstraction filter 150 creates explicit control, input and presentation statements relating to provided user description from the query interfaces 158, static UDO 140 and the dynamic UDO 144. This abstraction filter 150 further separates explicit personal user information from the adaptation engine 162, which may be externally situated with regard to the user local system 132 in some implementations.

The abstraction filter 150 classifies the user information into the following functional groups:

35

- Control (manipulation and activation of interface elements)
- Entry (e.g., data input)
- Presentation (output to the user)
- Authorization (privacy, security)

5 The Control Function includes the modes:

- Selection (e.g., mouse, voice, etc.)
- Navigation (e.g., mouse, arrow key, TAB key, voice, etc.)
- Activation (e.g., mouse, "enter" key, stylus, voice, etc..)

The Entry Function includes the function(s):

- 10 • Activation Mode (e.g., keyboard, switch, voice, mouse, stylus, etc.)

The Presentation Function includes the modes:

- Visual (e.g., monitor characteristics, GUI features, video features, etc.)
 - Aural (e.g., level, bandwidth, voice features, other sounds, etc.)
 - Tactile (e.g., texture range, features, etc.)
- 15 • Gestural (e.g., gesture language, recognition thresholds, etc.)
- Haptic (e.g., force levels, degrees of control, etc.)

The Authorization Function includes the modes:

- Identification
 - Role
- 20 • Authentication
- Permissions

The output of the abstraction filter 150 is sent to the preference object/capability object assembly 154.

10. Preference Object/Capability Object Assembly

25 The preference/capability assembly 154 generates the preference objects based upon the output of the abstraction filter 150. The preference/capability assembly 154 also either generates or retrieves a capability object that is associated with the pending session. The format of one embodiment of a preference object and a capability is described below in further detail in section II, entitled "Formal Object Syntax.". In other embodiments the capability object may be retrieved by the adaptation engine 162 or provided directly to the adaptation engine 162 by the user local system 132.

30 11. User's Proxy

The proxy 136 is an agent of the user 120 which may be employed when the user local system 132 cannot make the necessary changes in the user experience device 128 or the information source 166 on its own. For example, the user local system 132 may not have the capability of converting text to speech or speech to text (this would have been expressed in a capability object to the adaptation engine 162). The function of the user's proxy 136 is to execute 35 an adaptation object transmitted by the adaptation engine 162. Although the proxy 136 is an agent of the user, and is

shown in the user domain 104, its physical location could be in several different physical locations. It could be local, as a plug-in to a browser, for example, or it could be a remote service, located with an Internet service provider, an applications service provider, a Web site, in the information source, in an electronic device, or even in the adaptation engine.

5 12. Learning Component

The learning component 13 is an "artificial intelligence" device that is used to enhance and accelerate preference object generation. It collects adaptation result summaries from the adaptation engine 162 and correlates these with the preference objects and capability objects that were generated during a session. Over time, the learning component 13 generates new rules for preference object generation that are incorporated into the UDOs. In one
10 embodiment of the invention, these rules take priority over previous rules in the UDOs in considerations for generating the prototype preference objects, since they represent shortcuts in the preference object generation process.

13. Information Source

The information source 166 includes any information, service, resource, application, collaboration, interaction, whether external or internal to the user local system 132. The information source 166 includes, but is not
15 limited to, such things as databases, information bases, registries, repositories and other storage facilities, applications, agents, web sites, Internet Service Providers, Application Service Providers, chat rooms, collaborations and conferencing, devices and device drivers, adaptation and conversion services. For session management and preference object generation, the information source 166 is capable of providing to the adaptation engine 162 a reference identifier that is associated with the user 120 and also providing capability objects that describe its
20 capabilities for potential accommodation of the user's preferences and capabilities.

The information source 166 may include: a computer; a personal appliance; an ATM; a kiosk; a handheld device; a smart appliance, a network of devices, a networked federation of computers; a game system; electronic instrumentation; an automobile; a television; a telephone; a lamp; an air conditioning system; a sprinkler system; an elevator; or a monitoring and control system for a room, such as family room, an office, or an elevator. Furthermore, in
25 one embodiment of the invention, the information source 166 is integrated with the user local system 132.

In one embodiment of the invention the information source 166 includes one or more adaptable applications. An adaptable application provides an abstract user interface definition which is available to programmers. In another embodiment of the invention, the information source 166 is comprised of Extensible Style Sheets (XSL) and XSL translators (XSLT) recommended by the World Wide Web Consortium (W3C).

30

II. Formal Object Syntax

Set forth below is a formal description of the session objects, the preference objects, the capability objects, adaptation protocol objects, and adaptation objects. In the symbol descriptions below, **P** stands for any string of alphabetic characters and *n* stands for an integer. For example, P_n could stand for M_3 , CML_2 or P_7 .

35

- P_n : exactly one instance of $P_n \rightarrow$ the object is required;
- P_n^i : exactly i instances of $P_n \rightarrow i$ instances of the object are required;
- $P_n^?$: zero or one instance of $P_n \rightarrow$ the object occurs at most once;
- P_n^+ : one or more instance of $P_n \rightarrow$ the object should occur at least once;
- 5 P_n^* : zero or arbitrary instances of $P_n \rightarrow$ the object may occur any number of times;
- $P_n^{i,j}$: between i instances and j instances of $P_n \rightarrow$ the object may occur any number of times between i and j , inclusive ; when '&' is substituted for j , the object may occur an arbitrary number of times more than i .

10 These symbols will be used in a formal syntax to express the structure of objects. For example, as will be discussed more fully below, an exemplary description of a preference object is as follows.

- Preference Object $\rightarrow P_1 P_2^+ P_3$
- P_1 ID
- P_2 Preference Object Function
- 15 P_3 Adaptation Session

20 The foregoing description specifies that an object called a preference object is made up of three types of object, P_1 , P_2 and P_3 , and that there may be one or more occurrences of objects of type P_2 . The objects appearing in this syntax have referent names—e.g., ID, preference object Function and Adaptation Session—which are defined in similar syntax subsequent to this definition.

Such syntax can be used and rendered in many ways. In the Illustrative Example that appears farther on, this syntax is rendered in an XML-like form. Thus, the above definition might be rendered in usage as

```

25 < PREFERENCE OBJECT >
    < ID > </ID >
    < PREFERENCE OBJECT_FUNCTION > </PREFERENCE OBJECT_FUNCTION >
    < PREFERENCE OBJECT_FUNCTION > </PREFERENCE OBJECT_FUNCTION >
    < PREFERENCE OBJECT_FUNCTION > </PREFERENCE OBJECT_FUNCTION >
    .....
30 < PREFERENCE OBJECT_FUNCTION > </PREFERENCE OBJECT_FUNCTION >
    < ADAPTATION_SESSION > </ADAPTATION_SESSION >
</PREFERENCE OBJECT >
    
```

The formal mechanism is hierarchical and can extend downward until some explicit terminal symbols are defined, ending the extension. It is noted that some definitions may be supplied by recommendations of standards bodies, by equipment manufacturers, software developers and the like, or future developments.

Adaptation Session

5 A description of an adaptation session object is set forth below.

Adaptation Session → E₁ E₂⁺

E₁ ID

E₂ Adaptation Session Reference

10

The ID field of the Adaptation Session refers to the session to which a containing object belongs. The Reference field contains references to other concurrent sessions which may be related to this session, e.g., in a collaboration. This definition is used throughout the rest of the formal definitions, hence is referred to as "global."

15 Preference Object

A description of the of the preference object is set forth below.

Preference Object → P₁ P₂⁺

P₁ ID

20 P₂ Preference Object Function

ID → PI₁ PI₂ PI₃⁺

PI₁ Name

PI₂ Source

25 PI₃ Adaptation Session

The Name is a unique handle by which the preference object will be identified. The Source identifies the entity that generated the preference object. The Adaptation Session identifies the session with which the Preference Object is associated.

30

35

Preference Object Function → PF₁ PF₂ PF₃⁺ PF₄^{*} PF₅

PF₁ Type

PF₂ Name

PF₃ Mode

5 PF₄ External Reference

PF₅ Preference

Reference → PMR₁ PMR₂?

PMR₁ External Name

10 PMR₂ Locator

External Name → PEN₁ PEN₂

PEN₁ capability object Name

PEN₂ capability object Function

15

Preference → PMP₁ PMP₂ PMP₃⁺

PMP₁ Name

PMP₂ Priority

PMP₃ Characteristics

20

Characteristics → PMPC₁ PMPC₂

PMPC₁ Name

PMPC₂ Descriptor

25 The Type of the Preference Object Function identifies the type of the Preference Object Function, *e.g.*, Entry, Control, Presentation and Authorization. The Name field stores a unique handle that is used for correlating the Preference Object Function with Capability Object Functions and other entities. The Mode field identifies the mode of the Preference Object Function, *e.g.*, visual for presentation, activation for Entry, selection for Control. When a preference is identified, it should be in the context of a specific device, *e.g.*, a monitor. Thus, a component of the Reference object should point to the capability object where this device is described.

30

 The Preference Object contains a priority for the preference with respect to other preferences. Furthermore, the Preference Object may describe characteristics of the preference. The Name field of the Preference is used to refer to it within a selected session. The Descriptor field contained in the Characteristics object is not further defined here.

35

Capability Object

A description of the capability object is set forth below.

Capability Object → C₁ C₂⁺

C₁ ID

5 C₂ Capability Object Function

ID → CI₁ CI₂ CI₃⁺

CI₁ Name

CI₂ Source

10 CID₃ Adaptation Session

The Name field is a unique handle for the capability object. The Source field identifies the entity that generated the capability object; there may be multiple capability objects generated. Furthermore, the Source field identifies type of the capability object, i.e., a user capability object or an information source capability object. Since
 15 capability objects may be related to several concurrent sessions, the Adaptation Session field contains references to these.

Capability Object Function → CF₁ CF₂ CF₃⁺

CF₁ Type

20 CF₂ Name

CF₃ Capability

The Type object of the Capability Object Function contains one of the following types: (i) entry, (ii) control, (iii) presentation, (iv) authorization, (v) application, (vi) operating system, (vii) processor system, and (viii)
 25 communications.

The Name fields stores a unique handle that Preference Object Function fields and other entities can reference. The Capability field describes the specific characteristics of an interface entity, including devices, systems, programs, data sets, and graphical user interface elements. In some cases, particularly when the Type is a device, the contents of the Capability field may actually be provided externally, through the Reference, (see Capability definition,
 30 below). For example, rather than including all the capability information about a device in the capability object, the Reference field can point to a source for the manufacturer's description of the device's capabilities. The Reference field might also be used to cross-refer to other capability objects and possibly preference objects.

35

Capability → CP₁ CP₂⁺ CP₃^{*} CP₄⁺ CP₅⁺ CP₆[?]

CP₁ Type

CP₂ Reference

CP₃ Executables

5 CP₄ Characteristics

CP₅ Adaptation Object Listener

CP₆ Action

Characteristics → CCH₁ CCH₂⁺

10 CCH₁ Name

CCH₂ Descriptor

Adaptation Object Listener → CML₁ CML₂ CML₃

CML₁ ID

15 CML₂ Reference

CML₃ URI

The Type field of the Capability specifies the type of entity that the Capability refers to, *e.g.*, a mode for an interface function, when the Capability Object Function Type is one of Entry, Control, Presentation or Authorization, or
 20 a performance feature, such as throughput on the communications interface, when the Capability Object Function Type is Communications. The Executables field can contain classes that embody the capability, rather than a description of characteristics, which might not be complete enough or may be too complex to be usable. The Characteristics field contains the relevant capability descriptions of the entity. The Adaptation Object Listener field identifies an entity that is expecting to receive any adaptation object that is generated using the contents of the capability object. For
 25 example, the Adaptation Object Listener field might reference an application on the user local system that is expecting to be transformed in some way—an adaptable application. The proxy 136 is another example of a Adaptation Object Listener that could be designated. The Adaptation Object Listener field is associated with a capability, since more than one entity may be needed to act on an adaptation object. The Action field is used to direct the adaptation engine
 162 to take some action, *e.g.*, to “get” a set of capability characteristics from a previously cached capability object or from
 30 a remote source.

The ID field in the Adaptation Object Listener field is a unique handle for identifying the Adaptation Object Listener. The Reference field is a locator for the entity specified to be a Adaptation Object Listener associated with the capability object. This could be an internal address when adaptation takes place within a single system, or a remote reference. Optionally, an URI (Uniform Resource Identifier) can be used.

35

Adaptation Protocol Object

A description of the adaptation protocol object is set forth below.

Adaptation Protocol → MP₁, MP₂*

5 MP₁ ID

MP₂ Adaptation concept

ID → MPI₁ + MPI₂

MPI₁ Name

10 MPI₂ Session

The Name field is a unique handle for identifying the Adaptation Protocol. The Session field identifies a session that is associated with Adaptation Protocol .

15 Adaptation Concept → MT₁, MT₃*

MT₁ Type

MT₃ Adaptation Constraint

20 The Type object of the Adaptation Concept identifies the type of characteristic or specialization that the Adaptation Concept will be concerned with. Such characteristics will include, among many others, size, color, speed, screen layout, sound volume, contrast, and font family. In general, this type will be associated with one of the adaptation service units 228. The Adaptation Constraints field stores the limits on the characteristic(s) associated with the Adaptation Concept.

25 Adaptation Constraint → MTC₁, MTC₂

MTC₁ Constraint Description

MTC₂ Weight

30 Typical Constraint Descriptions include maximum, minimum and preferred values. However, not all characteristics can be limited in this manner. For example, with red-green colorblindness, colors should be restricted to blue-yellow. However, the maximum blue is white and the minimum blue is black, and similarly for yellow. Moreover, some blue-greens are considered to be blues, may even contain red components, and still be distinguishable by a person who is red-green colorblind. The Weight object is a set of values that could be used as parameters in the adaptation engine learning and decision functions, such as neural nets and genetic algorithms.

35

Adaptation Object

A description of one embodiment of an adaptation object is set forth below.

Adaptation Object $\rightarrow M_1 + M_2^* + M_3? + M_4^*$

- 5 M_1 ID
 M_2 Adaptation Function
 M_3 Configuration
 M_4 Setup
- 10 ID \rightarrow MID₁ MID₂
MID₁ Name
MID₂ Adaptation Session

Although there is at most one Configuration field associated with an Adaptation Object, there can be more than one Setup field, since the configuration may be complex, with possibly multiple protocols being used among multiple communicating entities. Each Setup field references a particular entity in the Configuration field. In general, the Configuration field and its associated Setup field will be used only when there are different components and interconnections than in the initial arrangement. These might occur in dynamic situations such as during collaborations, conferences or federations (e.g., chat rooms, auctions), where entities are entering and leaving the active group.

Adaptation Function $\rightarrow MF_1 MF_2^* MF_3 MF_4^*$

- MF_1 Type
 MF_2 Executables
25 MF_3 Specification
 MF_4 External Reference

The Type field of the Adaptation Function indicates the type of function that is being represented by the Adaptation Function, *i.e.*, Entry, Control, Presentation, and Authorization. The Executables field contains classes that may be needed to execute the intended transformations. For example, such classes might be used to carry out rendering of visual objects on a monitor screen. The Specification field specifies how the transformations are to take place. In one embodiment of the invention, the Specification field contains script or code that describes the use of the classes in the Executables field. The External References field contains references to external resources that might be needed to execute the transformations specified. For example, the External References field might contain a reference to a metadata file.

Configuration → MC₁ MC₂^{2&} MC₃⁺

MC₁ ID

MC₂ Component Name

MC₃ Topology

5

The Configuration field carries an identifier ID since the execution of the configuration may be done by a different entity than the one carrying out the Adaptation Functions. The Components Name field comprises of those entities which will participate in the interaction, listed by identifier (e.g., handle and IP address). In collaborations, for example, the Components Name field lists all parties in the collaboration. The Topology field identifies the directed graph describing the interconnections among the members of the Components field.

10

Topology → MTC₁¹

MTC₁ Entity Pair

15 Entity Pair → MCTE₁ MCTE₂² MCTE₃

MCTE₁ Name

MCTE₂ Entity

MCTE₃ Interaction

20 The Entity field is a Component Name in the Configuration field. The Interaction field identifies an entity-level protocol with which the entities carry their interaction.

Setup → MS₁ MS₂ MS₃⁺ MS₄^{*}

MS₁ ID

25 MS₂ Pair Reference

MS₃ Protocol

MS₄ Executables

30 The Setup field carries an identifier ID since the execution of the setup may be done by a different entity than the one carrying out the Adaptation Functions. The Locator object refers to a pair of entities listed in a Components field of the Configuration field. The Protocol field identifies the communications-level protocol that will be used by the two entities during the session, since sometimes two communicating entities may use more than one protocol to interact. There is also the possibility that one of the entities does not implement a protocol that is needed for interaction. In this case, classes for the protocol could be carried in the Executables field.

35

Protocol → MSP₁ MSP₂* MP₃*

MSP₁ Type

MSP₂ Port

MSP₃ Executables

5

The Type of the protocol field describes the type of protocol to be used (*e.g.*, transport, file transfer, streaming, hypertext, etc.), by name (*e.g.*, TCP/IP for transport, FTP for file transfer, HTTP for hypertext). The Port field identifies a port where the protocol will be accessed, *e.g.*, port 80 for HTTP. The Executables object contains a set of classes that implement the protocol, when needed.

10

III. Method of Operation

Figure 3 is a flowchart illustrating one embodiment of the adaptation process of the adaptation system 100. It is to be appreciated by a skilled technologist that depending on the embodiment, selected acts shown in the flowchart may be omitted and that others may be added. Furthermore, depending on the embodiment, the ordering of the acts may be varied.

15

Starting at a state 304, the user 120 accesses an electronic device, such the user local system 132. Next, at a state 308, the adaptation system 100 retrieves a user capability object that is associated with the accessed electronic device. In one embodiment of the invention, the accessed electronic device is requested to provide the user capability object. In another embodiment of the invention, the capability object is retrieved from a central depository of capability objects. In yet another embodiment of the invention, the capability object is retrieved from a server that

20

is maintained by the manufacturer of the accessed electronic device.

25

Next, at a state 312, the preference object for the user is retrieved. In one embodiment of the invention, the preference object is generated in real time in response to the user accessing the electronic device. In this embodiment, the adaptation system 162 generates the preference object based at least in part on either: (i) preferences specified by the user, or (ii) detected operating context conditions of the user 120 or the user local system 108. The operating context conditions can include: the proximity of the user 120 to the user local system 132, the ambient temperature, weather conditions, the level of background noise surround the user 120 or the user local system 132, etc.

30

Continuing to a state 316, the user 120 requests to use the information source 166. Proceeding to a state 320, the information source capability object that is associated with the information source 166 is retrieved. Moving to a state 324, the adaptation system 100 generates an adaptation object based at least in part upon the contents of the preference object, the user capability object, and the information source capability object.

35

Next, at a state 324 the adaptation system 100 uses the adaptation engine 162 to adapt the human factor information that is transmitted by the accessed electronic device and/or to adapt the control information that is received by the accessed electronic device.

Figures 4A-4D are collectively a flowchart illustrating in further detail one embodiment of the adaptation process shown generally by Figure 3. It is to be appreciated that depending on the embodiment, selected acts shown

40

by the flowchart may be omitted and that others may be added. Furthermore, depending on the embodiment, the ordering of the acts may be varied.

Starting at a state 400, the user 120 provides his or her interface preferences to the user input description module 124. Continuing to a state 404, the user input description module 128 transmits the gathered information to the static UDO 142. Moving to a state 408, the user 120 requests access to the user local system 132. Next, at a state 412, the user local system 132 notifies the user input description module 124 of the pending session.

Proceeding to a state 416, the user input description module 124 provides dynamic user information to the dynamic UDO 144. The dynamic user information can include: the proximity of the user 120 to the user local system 132, the ambient temperature, weather conditions, the level of background noise surround the user 120 or the user local system 132, etc. Next, at a state 420, the dynamic UDO generates UDO objects based upon the provided information.

Moving to a state 428, the user local system 132 contacts the information source 166 to begin a session. Continuing to a state 432, the accessed information source provides a capability object to the adaptation engine 162. Next, at a state 436, in one embodiment, the adaptation engine 436 requests the query interfaces 158 to provide a user capability object and a preference object. In another embodiment the capability object may be retrieved by the adaptation engine 162 or provided directly by the user local system. At the state 440, the query interface 158 invokes the static UDO 142 and the dynamic UDO 144 to provide their respective UDO objects that are associated with the pending session to the abstraction filter 150.

Continuing to a state 444, the UDO objects are received by the abstraction filter 150. At a state 444, the abstraction filter 150 converts the contents of the static UDO objects and the dynamic UDO objects into interface function descriptions. The interface function descriptions are explicit statements describing how a particular aspect of the user interface is to be changed, *e.g.*, larger font size, or render text aurally.

Next, at a state 452, the abstraction filter 150, transmits the interface function descriptions to the preference/capability assembly 154. Moving to a state 456, the preference/capability assembly 154 generates the preference object and retrieves the user capability objects. In one embodiment of the invention, the capability object is generated using information that is transmitted by the user local system 132.

Proceeding to a state 460, the preference/capability assembly 154 transmits the preference objects and the user capability object to the adaptation engine 162. Moving to a state 464, the adaptation engine 162 creates an adaptation object based at least in part upon content of the preference object, the user capability object, and the information source capability object. The process of generating the adaptation object is described in greater detail below with respect to Figure 5.

Next, at a state 468, the adaptation engine 162 transmits, depending on the content of the adaptation object, the adaptation object to either the proxy 136 or the user local system 132.

In one embodiment, all adaptation objects are sent by default to the user local system 132. However, if the user local system 132 cannot perform adaptation specified by the adaptation object, the adaptation object is

transmitted to the proxy 136. Continuing to a state 472, the adaptation object is used to transform the human factor information that is transmitted to the user 120 or, alternatively, the control information that is received by the client with respect to the accessed electronic device or the information source 166.

5 Figures 5A and 5B are collectively a flowchart illustrating a process that is performed by one embodiment of the adaptation engine 162. Figures 5A and 5B illustrate in further detail the acts that occur in state 464 of Figure 4. It is to be appreciated that depending on the embodiment, selected acts shown by the flowchart may be omitted and that others may be added. Furthermore, depending on the embodiment, the ordering of the acts may be varied.

10 In the states 500, 504, and 508, the preference object/capability object registry 208 respectively receives the information sources capability object, the user capability object, and the preference object(s) through the communications interface 204.

Next at a state 512, the adaptation engine 162 caches the user capability object and information sources capability object in the capability cache 216 (Figure 2). Continuing to a state 516, the adaptation engine 162 caches the preference object in the preference cache 212. Next, at a state 520, the adaptation event manager 220 starts a session with the session manager 248 and invokes the Adaptation Manager 224. Furthermore, the adaptation engine 15 162 starts logging events in the accounting log 252. Events are continuously added to this logging file throughout the adaptation process.

Continuing to a state 524, the adaptation manager 224 examines the preference object(s) and capability objects associated with the pending session in the preference object cache 212 and capability object cache 216, respectively, and selects one of the adaptation service units 228, based upon a stored rule base and preference 20 object/capability object information.

Next, at a state 528, output of the adapter service units 228 is collected to form an adaptation object skeleton having a plurality of adaptation concepts. Continuing to a state 532, the interpretive consolidator 232 optimizes the adaptation skeleton by identifying conflicts between the various concepts. The interpretive consolidator 232 also integrates adapter service unit results to be consistent with other knowledge the interpretive consolidator 25 232 may have about the ongoing transaction and previous transactions with this or other users. For example, the output of one adapter service unit may specify that since a display that is in use by the user 120 is a black and white monitor, all output to the user on a display should be in shades of 8-bit gray scale. However, the output of another one of the adapter service units may specify that, that all color green should be converted to orange, since the user has stated a preference for it, whether by explicit choice or implicitly, *e.g.*, from colorblindness. If the two rules resulted in 30 a poor presentation, the interpretive consolidator 232 would correct the conflict.

Next, at a state 536, the adaptation object assembler 240 collaborates with any adaptation service that has been specified by the capability or preference objects to generate an adaptation object. Moving to a state 540, the adaptation engine 162 transmits the adaptation object to the communications interface 204 for transmission either to the proxy 136 or the user local system 132. The process flow then returns to state 468 of Figure 4.

35

IV. Illustrative Examples

The following illustrates one exemplary function of the adaptation system 100. It is noted that the adaptation system 100 can be used in a multitude of other contexts and be embodied in other electronic devices than those that are described below.

5 Jennifer Adams uses a wheelchair for mobility since she lost the use of all extremities in a diving accident. She uses a personal accessor (the user local system 132), a mobile electronic device, mounted on her wheelchair which she enters data and controls using her voice. The accessor has an 8x6 screen. The operating system and graphical user interface (GUI) of the applications installed on the accessor display Jennifer's personal look-and-feel. The accessor has an RF device for coupling to various external devices. There is a discovery service running over the RF
10 link for identifying, locating and connecting to the external devices or systems. The RF system also has a means for determining how far away the accessor is from a device it is to be coupled with, *e.g.*, signal strength.

When Jennifer comes close to a system or device she wants to interact with--an ATM (the information source 166), for example--the RF system detects a signal from the ATM and the accessor executes a handshake with the ATM, identifying itself and Jennifer to the ATM. As part of this process, the accessor provides a reference to its
15 characteristics in case they are needed, and a reference to Jennifer's persistent preferences. When the preference data is retrieved, it can be determined that control and entry functions are not to be operated using hands and that voice is preferred for these modes of interaction. The accessor has a means of uniquely identifying Jennifer (such as via a retinal scan or voice print) and her PIN is sent to the ATM without her having to speak it, to avoid revealing the PIN to anyone who might be listening.

20 As Jennifer gets closer to the ATM, she notices that she will not be able to see the ATM screen because the sun is glaring on it and she is unable to maneuver her wheelchair in such a manner as to compensate for the glare. She speaks to the accessor: "New preference: send visual display here." The accessor sends this to the ATM along with a reference to the accessor's visual display characteristics (if this reference was not already sent in the handshake). The ATM uses this information to transform the screen content to comply with the accessor screen and the GUI of the
25 application used to handle bank transactions. The transformation takes place and Jennifer is able to carry out her transaction on her accessor.

In the scenario above, the ATM is assumed to be an adaptable application, as is the accessor device. The latter need not be an adaptable application in all cases. The accessor is the user local system 132 while the ATM is the information source 166. It is also assumed that there is a secure protocol between the accessor and ATM,
30 referred to in the sequel as the 'ATMSecureRF', which carries the monetary transactions between the accessor and ATM.

When the ATM system initializes (perhaps just after installation), it generates a capability object. This capability object, which specifies the adaptable capabilities of the system, is registered with a designated adaptation engine, which may be co-located with or part of the ATM or could be a service of the bank that owns the ATM. The

partial contents of the capability object appear as below (expressed in an XML-like syntax for illustrative purposes). In one embodiment, the capability object is as follows.

```

< CAPABILITY OBJECT ID="ATM Interface Capability" SRC="New Union ATM"
  ADAPTATION_SESSION=null >
5   .....

  < FUNCTION TYPE="control" >
    < CAPABILITY MODE="selection" ADAPTATION OBJECT_LISTENER="touchscreen" >
      < CHARACTERISTIC NAME="input source" >
10      < DESCRIPTOR>"ATMSecureRF" </DESCRIPTOR>
      </CHARACTERISTIC >
    </CAPABILITY >
    < CAPABILITY MODE="activation" ADAPTATION OBJECT_LISTENER="touchscreen" >
      < CHARACTERISTIC NAME="input source" >
15      < DESCRIPTOR>"ATMSecureRF" </DESCRIPTOR>
      </CHARACTERISTIC >
    </CF_CAPABILITY >
  < FUNCTION TYPE="entry" >
    < CAPABILITY TYPE="activation" ADAPTATION OBJECT_LISTENER="touchscreen" >
20    < CHARACTERISTIC NAME="input source" >
      < DESCRIPTOR>"ATMSecureRF" </DESCRIPTOR>
      </CHARACTERISTIC >
    </CAPABILITY >
  < FUNCTION TYPE='presentation' >
    < CAPABILITY TYPE="visual" ADAPTATION OBJECT_LISTENER="touchscreen" >
25    < CHARACTERISTIC NAME="output sink" >
      < DESCRIPTOR>"ATMSecureRF" </DESCRIPTOR>
      </CHARACTERISTIC >
    </CAPABILITY >
30  </FUNCTION >
  .....

</CAPABILITY OBJECT >

```

The foregoing capability object will have an adaptation session identifier assigned each time that a new session begins, so that it carries a null value when first registered with the adaptation engine. The capability object shows that the ATM has a touchscreen for both Control and Entry, indicated by the Adaptation Object Listener field, but that input can be accepted from an ATMSecureRF channel, and that the touchscreen display can also be sent externally over an ATMSecureRF channel.

When Jennifer's accessor device comes within range of the ATM, the ATM initiates the establishment of a secure communications channel using the ATMSecureRF protocol. After establishment of the secure channel, Jennifer's accessor initiates an identification sequence, analogous to the insertion of a physical ATM card into the machine followed by transmission of a personal identification number (PIN). This sequence also includes transmission of a unique authentication, perhaps a digital signature, that will be used to obtain information from Jennifer's user description object. This authentication is generated anew each time the accessor makes a new connection with a system or device. Channel establishment and the identification sequence are a part of the ATMSecureRF protocol and as such are outside the scope of this invention.

Upon the verification of Jennifer's identification, the ATM generates a preference object. The preference object, containing Jennifer's unique identification, is sent to the adaptation engine. The information contained in the preference object is as follows:

```

< PREFERENCE OBJECT ID="ID preference object" SRC="NewUnion ATM" ADAPTATION_SESSION="12345">
  < FUNCTION TYPE="authorization">
    < MODE TYPE="identification" NAME="user-handle">
      JenniferAdams@uapcoalition.org
    </MODE>
    < MODE TYPE="authentication" NAME="userid">
      [Jennifer's digital signature]
    </MODE>
  </FUNCTION>
</PREFERENCE OBJECT>

```

The ATM assigns an adaptation session number and also passes this number to the accessor for reference. The "user-handle" indicates to the adaptation engine where to find Jennifer's user description object (UDO).

The adaptation engine 162 pairs the ATM registered capability object with this preference object by assigning the adaptation session number, e.g., "12345", to a copy of the capability object. This pairing triggers the adaptation sequence. The first step in the adaptation sequence involves obtaining a preference object for Jennifer's situation at establishment of the session. The adaptation engine 162 passes the authentication from the preference object to the query interface 158, which verifies Jennifer's identity.

Section of Jennifer's static UDO (Secured Information)

< FACULTY TYPE="mobility" >
 < FUNCTION TYPE="ground" STATUS="assisted" > wheelchair < /FUNCTION >
 5 < FUNCTION TYPE="stair" STATUS="none" > < /FUNCTION >
 < /FACULTY >

< FACULTY TYPE="motor" >
 < FUNCTION TYPE="hand" STATUS="none" > bilateral < /FUNCTION >
 10 < FUNCTION TYPE="leg" STATUS="none" > bilateral < /FUNCTION >
 < /FACULTY >

< FACULTY TYPE="touch" >
 < FUNCTION TYPE="hand" STATUS="none" > bilateral < /FUNCTION >
 15 < /FACULTY >

< PREFERENCE TYPE="entry" LOCUS="accessor" PRIORITY="0" >
 voice
 < /PREFERENCE >
 20 < PREFERENCE TYPE="control" LOCUS="accessor" PRIORITY="0" >
 voice
 < /PREFERENCE >
 < PREFERENCE TYPE="authorize" LOCUS="accessor" PRIORITY="0" >
 voice
 25 < /PREFERENCE >
 < PREFERENCE TYPE="presentation" LOCUS="accessor" PRIORITY="0" >
 < VISUAL DEVICE=[Acme reference] UI=[GUI reference] / >
 < /PREFERENCE >
 < PREFERENCE TYPE="presentation" LOCUS="accessor" PRIORITY="1" >
 30 < AURAL DEVICE=[Acme reference] API=[sound API reference] / >
 < /PREFERENCE >

< PREFERENCE TYPE="entry" LOCUS="extra" PRIORITY="0" >
 accessor: ATMSecureRF;
 35

```

    </PREFERENCE >
    < PREFERENCE TYPE="control" LOCUS="extra" PRIORITY="0" >
        accessor: ATMSecureRF;
        .....
5    </PREFERENCE >
    < PREFERENCE TYPE="authorize" LOCUS="extra" PRIORITY="0" >
        accessor: ATMSecureRF;
        .....
10   </PREFERENCE >
    < PREFERENCE TYPE="presentation" LOCUS="extra" PRIORITY="0" >
        visual: default;
        aural: default;
    </PREF >

```

15 The structure of the static UDO is such that any faculty which is not impaired (presumably for some time that is "long", e.g., compared to a mean session length) does not have an entry. Thus, Jennifer's vision, which is not impaired, does not appear in an entry in the UDO.

 Jennifer's UDO content indicates that she prefers interacting with her accessor through voice for Entry and Control functions and visually for Presentation functions. As a second choice, she will accept aural presentations from
20 her accessor, in case she is not able to see its screen for some reason. Whenever she is interacting with an entity—a system or device—beyond her accessor, she prefers (PRIORITY="0") to use the accessor for Entry and Control inputs to the entity and the default Presentation functions of the entity, indicated by visual: default and aural: default.

 Since the ATM is an entity beyond the accessor (LOCUS="extra"), the preferences corresponding to this are picked up for inclusion in the preference object. The abstraction filter 150 first deduces from the UDO that Jennifer
25 has no basic visual disabilities, since there is no FACULTY entry for vision in her static UDO. It also sees that her physical condition does not constrain head movement (only her arms, hand and legs are affected), so her field of vision is not going to be limited by not being able to move her head. [Although this is not an issue here, if Jennifer's head were restricted in its movement and her effective field of view thereby constrained, this would have to be taken into account in the accommodation, perhaps through a field-of-view parameter.] Consequently, the preference for using
30 the default visual presentation can be used without any limitation. The other interface functions of Entry and Control will be mediated through the accessor over an ATMSecureRF channel, among other possible means of interconnecting.

 This information, along with a reference to accessor capabilities maintained by the manufacturer of Jennifer's accessor, is passed to the preference/capability assembly 154, where a new preference object and a capability object are created. These are identified by the adaptation session number previously sent by the ATM to the
35 accessor and by unique names. The accessor also assigns itself a unique handle to identify itself in the adaptation

session. The ATM will assign an ATMSecureRF port address to accompany the accessor's name. For the example, the name can be "jadams". These are sent on to the adaptation engine 162, where they are analyzed by the adaptation manager 224 and found to be consistent with the capability object previously registered by the ATM. This results in an adaptation object that essentially verifies this consistency and provides the necessary setup for the transaction interaction.

```

5  < ADAPTATION OBJECT ADAPTATION_SESSION="12345" >
      < FUNCTION TYPE="control" MODE="selection" ID="transaction" >
          remote: jadams;
10  < / FUNCTION >
      < FUNCTION TYPE="control" MODE="activation" ID="transaction" >
          remote: jadams;
      < / FUNCTION >
      < FUNCTION TYPE="entry" MODE="activation" ID="transaction" >
15  remote: jadams;
      < / FUNCTION >
      < FUNCTION TYPE="presentation" MODE="visual" ID="transaction" >
          local: NewUnion ATM;
      < / FUNCTION >
20  < CONFIGURATION ID="transaction" >
      < COMPONENTS >
          "NewUnion ATM";
          "jadams";
      < /COMPONENTS >
25  < TOPOLOGY >
      < PAIR NAME="pair1" >
          "NewUnion ATM";
          "jadams";
          ATMSecureRF/ATMtransaction
30  < /PAIR >
      < /TOPOLOGY >
      < /CONFIGURATION >
      < SETUP NAME="transaction" >
          < PROTOCOL FOR="pair1" TYPE="ATMSecureRF" >
35  port: 88;

```

```

                speed: default;
            </PROTOCOL >
        </SETUP >
    </ADAPTATION OBJECT >

```

5

The foregoing adaptation object is essentially a confirmation of the mapping from the ATM's touchscreen inputs to the accessor via the ATMSecureRF channel, using a protocol called ATMtransaction, which will allow Jennifer to use her voice commands to the accessor to provide the transaction inputs. The ID="transaction" entries on the Function, Configuration and Setup tie all these fields together. This adaptation object is sent to the ATM, which then carries out the mapping. Information on the configuration and setup are sent to the accessor as well and the ATM notifies the accessor to begin the transaction using the ATMSecureRF.

Continuing the example, Jennifer realizes that she needs to get the presentation displayed on the accessor screen as well. The accessor transaction application is now running, controlled by her voice. There is also an application with which she remains in control of the accessor, separate from the transaction application. Jennifer uses this application to indicate her desire to redirect the ATM's presentation to the accessor, and a preference object is generated. The preference object, sent earlier, caused a capability object from the accessor manufacturer to be accessed and cached in the adaptation engine, with a session number. This same session number is used in the new preference object to cross-reference it with the accessor capability object

Jennifer's vision is virtually impaired, temporarily, by the sun angle and her position. This is entered into her dynamic UDO, maintained on her accessor, as

```

    < FACULTY TYPE="vision" SESSION="12345" >
        < FUNCTION TYPE="all" STATUS="not available" LOCUS="extra" >
    </FACULTY >

```

25

This indicates to the accessor that Jennifer cannot see something beyond the accessor ("extra") but that the accessor itself is still visible. The storage of this item in the dynamic UDO is triggered by Jennifer's command to the accessor to have the ATM direct its visual display to the accessor. A constraint message will be forwarded to the static UDO repository to be combined with Jennifer's information there to form a preference object and the necessary capability objects to complete the preference transaction. The constraint message is cached in the dynamic UDO at least until the session ends and thereafter on a least-recently-used caching discipline.

Because the accessor is a standard device, its characteristics and capabilities are referenced indirectly and have been imported by the adaptation engine 162. The preference object indicates the new preference of local display of the ATM transaction presentation:

35

```

5  < PREFERENCE OBJECT ID="jadamsP1" SRC="jadams" ADAPTATION_SESSION="12345" >
    < FUNCTION TYPE="authorization" >
        < MODE TYPE="identification" NAME="user-handle" >
            JenniferAdams@uapcoalition.org
        < /MODE >
        < MODE TYPE="authentication" NAME="userid" >
            [Jennifer's digital signature]
        < /MODE > < /MODE >
    < /FUNCTION >
10  < FUNCTION TYPE="presentation" >
        < MODE TYPE="visual" >
            < PREFERENCE TYPE="update" >
                < LOCUS APPLY="transaction" >
                    target: jadams;
15                    source: NewUnion ATM;
                    protocol: ATMSecureRF/ATMtransaction;
                < /LOCUS >
            < /PREFERENCE >
            < PREFERENCE TYPE="supplemental" >
20                capability object: jadamsC1;
            < /PREFERENCE >
        < /MODE >
    < /FUNCTION >
    < /PREFERENCE OBJECT >
25

```

This preference object repeats the identification information and adds a preference for the visual presentation to be delivered to the accessor, using the transaction setup already in place. The Preference shows a tag not defined formally earlier, the LOCUS tag. This will eventually indicate where the visual presentation will arrive for display. The "supplemental" preference points to a capability object, described below, containing additional

information regarding the accessor and Jennifer's preferred GUI.

```

35  < CAPABILITY OBJECT ID="jadamsC1" REF="jadamsP1" >
    < FUNCTION TYPE="authorization" >
        < MODE TYPE="identification" NAME="user-handle" >
            JenniferAdams@uapcoalition.org

```

```

    </MODE >
    < MODE TYPE="authentication" NAME="userid" >
        [Jennifer's digital signature]
    </MODE > </ MODE >
5 </ FUNCTION >
    < CAPABILITY TYPE="accessor" REF=[accessor mfr reference] ACTION="get"
        ADAPTATION OBJECT_LISTENER="jadams">
        characteristic: screen size;
        characteristic: screen resolution;
10 < LISTENER NAME="jadams" / >
    </CAPABILITY >
    < CAPABILITY TYPE="gui" REF=[GUI reference] ACTION="get"
        ADAPTATION OBJECT_LISTENER="jadams">
        characteristic: display width;
15 characteristic: display height;
        characteristic: font size;
        characteristic: font color;
        characteristic: minimum object width;
        characteristic: minimum object height;
20 .....(and so on).....

    </CAPABILITY >
    </CAPABILITY OBJECT >
25

```

This capability object makes reference to the accessor's characteristics, which the adaptation engine 162 should have already cached from a previous retrieval, and to the GUI that Jennifer uses. Some of the GUI characteristics are in fact preferences that Jennifer has specified, but are considered to be capabilities because she does not permit altering them. Note also that the preference object "jadamsP1", above, is cross-referenced.

The ATM has previously provided the adaptation engine 162 with a capability object that describes the visual interface of the ATM. The adaptation engine 162 uses this interface description and the accessor GUI description to render the ATM display for use on the accessor. This assumes that there is a generic, or abstract, description of the ATM visual interface available. In the case where there is not such a description, the ATM and accessor can revert to another standard interface.

The preference object and capability object above are transferred to the adaptation engine 162, where the adaptation procedure begins with the adaptation manager 224 examining the preference object and the capability objects to determine which adaptation service units 228 should be assigned. This can be done by examining the attributes in the tags of the preference object and capability objects. For example, the following characteristics taken
 5 from the accessor's capability information indicate that size and resolution of the accessor's display screen will be involved.

```

    < CHARACTERISTIC TYPE="screen" VALUE="size" UNIT="inches" >
      width: 8;
    10     height: 6;
    < / CHARACTERISTIC >
    < CHARACTERISTIC TYPE="screen" VALUE="resolution" UNIT="pixels" >
      width: 640
      height: 480;
    15     < / CHARACTERISTIC >
  
```

Thus, an adaptation service unit that can deal with display scaling is selected. The adaptation service unit uses the preference object and capability object information to begin building adaptation protocols, which are made up of adaptation concepts and adaptation constraints. Since the adaptation manager 224 does not yet have access to
 20 the full description of the ATM's current interface, the adaptation service unit can only build "approaches" and "outlines" of what is to be done—the adaptation concepts and constraints. Further detailing to build concrete adaptation objects will be provided by the adaptation service, assumed to have access to the description of the ATM's current interface, or by the interpretive consolidator 232. The adaptation concepts and constraints produced by the adaptation service units are collected into adaptation protocols which are then presented to the interpretive
 25 consolidator 232 for coordination and resolution of conflicts, among other services that it can provide.

Two adaptation protocols that can be produced using the preference object and capability object information cited above are as follows:

```

    < CONCEPT NAME="scaleGUI" >
    30     < CONSTRAINT TYPE="width" WEIGHT="0.7" UNIT="pixels" >
      preferred: 620;
      < /CONSTRAINT >
      < CONSTRAINT TYPE="aspect" WEIGHT="0.7" UNIT="none" >
      preferred: "maintain";
    35     < /CONSTRAINT >
  
```

```

5      < CONSTRAINT TYPE="font-size" WEIGHT="0.7" UNIT="pt" >
          min: 10;
          preferred: "scalewithGUI";
      < /CONSTRAINT >
5  < /CONCEPT >

      < CONCEPT NAME="scaleDisplay" >
          < CONSTRAINT TYPE="width" WEIGHT="0.7" UNIT="pixels" >
              preferred: 640;
10      < /CONSTRAINT >
          < CONSTRAINT TYPE="height" WEIGHT="0.7" UNIT="pixels" >
              preferred: 480;
          < /CONSTRAINT >
          CONSTRAINT TYPE="screen-width" WEIGHT="0.9" UNIT="inches" >
15      preferred: 8;
          < /CONSTRAINT >
          CONSTRAINT TYPE="screen-height" WEIGHT="0.9" UNIT="inches" >
              Preferred: 6;
          < /CONSTRAINT >
20  < /CONCEPT >

```

An adaptation protocol is constructed using the Adaptation concepts and constraints:

```

25  < M_PROTOCOL NAME="scale" SESSION="12345" >
          < CONCEPT NAME="scaleGUI" >
              [constraints]
          < /CONCEPT >
          < CONCEPT NAME="scaleDisplay" >
              [constraints]
30  < /CONCEPT >
          ....more concepts....
      < /M_PROTOCOL >

```

There can be several adaptation protocols generated, depending upon how complex the requested
35 transformation is. The interpretive consolidator 232 is called upon to coordinate these adaptation protocols and to

resolve any conflicts detected. It can also bring into the activity any knowledge that it might have about previous adaptation transactions that might have been requested by Jennifer during her session and even those of other people who might have interacted with the ATM prior to Jennifer. Because the example is fragmentary for tutorial purposes, there are no apparent conflicts to resolve. However, the interpretive consolidator 232 already has some knowledge about the ATM's display from a capability object that was submitted on its behalf:

```

5      < CHARACTERISTIC TYPE="display" VALUE="size" UNIT="pixels" >
          width: 800;
          height: 667;
10     < / CHARACTERISTIC >

```

The ATM's display has a different aspect angle than the GUI on Jennifer's accessor. If a direct mapping is made, the ATM's display on the accessor will appear to be "crushed". Consequently, the interpretive consolidator 232 proposes an additional Adaptation concept to account for the aspect angle differences:

```

15     < M_PROTOCOL NAME="scale" SESSION="12345" >
          .....
          < CONCEPT NAME="scaleFix" APPLY="scaleGUI" >
              < CONSTRAINT TYPE="width" WEIGHT="0.8" UNIT="pixels" >
20                 preferred: 585;
              < /CONSTRAINT >
              < CONSTRAINT TYPE="height" WEIGHT="0.8" UNIT="pixels" >
                  preferred: 480;
              < /CONSTRAINT >
25     < /CONCEPT >
          .....
        < /M_PROTOCOL >

```

The APPLY attribute in the CONCEPT tag indicates the Adaptation concept within the same Adaptation protocol to which the adjustment will apply.

Another conflict that will often arise is the scaling of objects that appear on the screen. When scaling down an object, the resulting rendering could be too small to adequately see or interact with. If objects are not scaled down enough, the display could appear too crowded. Consequently, some compromises should be considered. Often there are objects which are not necessary to the operation of the application or use of the content. For example, advertisements, while important for business reasons quite often have nothing to do with the intended activity. These

can be removed from the scaled display to preserve screen "real-estate." Another compromise is to use scrolling on a window, if the GUI permits this, although many users find this annoying. A competent Interpretive Consolidator will have many such compromises, that it can suggest to a Adaptation Service, built into its rules.

After all conflicts have be addressed and all enhancements and "tuning" have been applied to the adaptation protocols, they are sent, along with the preference objects and capability objects to an adaptation service, which will compute the adaptation objects required to concisely describe the transformations.

The tasks that the adaptation object generated should accomplish include the following, not all of which were discussed above:

- Provide a method to scale the display area of the ATM screen to that of the accessor.
- Provide a method to scale the application area of the ATM screen to fit the area allotted by the accessor GUI and a resident application (*e.g.*, a continuous display keeping Jennifer informed about where she is and what she is connected to).
- Provide a method to scale and render the ATM GUI objects to the accessor GUI, possibly eliminating some objects not critical to the operation of the ATM.

These processes are all algorithmic. It might be the case that it is more effective to send the algorithms to the accessor than to send the scaled display, because of bandwidth and latency on the connection between the accessor and ATM. Consequently, these can be executables. This decision can be made on the basis of the capability objects regarding the accessor platform, including operating system and languages that it can handle, communications bandwidth and processor capability. It is likely that scaling will be a common task, so that the algorithms may already be on the accessor; this would be indicated in a capability object. In this case, the adaptation object need only call out the algorithms by name and provide the parameters to go into them. The third task involves re-creating the ATM's display.

The adaptation object for this set of tasks might then appear:

```

25     < ADAPTATION OBJECT ADAPTATION_SESSION="12345" >
        < FUNCTION TYPE="presentation" ID="transaction" >
            < EXECUTABLE TARGET="jadams" REF="scaleScreen()" >
                start-width: 800;
                end-width: 585;
30                start-height: 667;
                end-height: 480;
            < /EXECUTABLE >
            < EXECUTABLE TARGET="jadams" REF="scaleAvailableArea()" >
                start-width: 760;
35                end-width: 500;

```

```

        start-height: 650;
        end-height: 427;
    </EXECUTABLE >
    < EXECUTABLE TARGET = "jadams" REF = "scaleObjects()" >
5         min: 30 pixels;
    </EXECUTABLE >
    < EXECUTABLE TARGET = "jadams" REF = "mapObjects()" >
        layout: layoutManager();
        events: eventSet;
10         .....
    </EXECUTABLE >
    </FUNCTION >
    < FUNCTION TYPE = "presentation" MODE = "visual" ID = "transaction" >
        remote: jadams;
15    </ FUNCTION >
    < CONFIGURATION ID = "transaction" >
        repeat;
    </CONFIGURATION >
    < SETUP NAME = "transaction" >
20         repeat;
    </SETUP >
</ADAPTATION OBJECT >

```

25 Note that reference is made to "transaction" throughout. This reference links the adaptation object to the previous adaptation object which initialized the interaction between the accessor and the ATM. The presentation is redirected to the accessor and the relevant changes to be made to the style of the presentation are indicated by the executables. The adaptation object is sent to both the ATM and the accessor. The ATM ignores the executables because of the TARGET attribute in the EXECUTABLES tag.

Jennifer can now interact with the ATM and see the results of her interactions on the accessor's screen.

30 Because each user has different needs from other users, and systems and collections of devices may have configurations that change with every session, the present system uses "self-descriptions" of information system to provide flexibility. This information is used to determine operating conditions and configurations for a given session, and to support dynamic changes in these conditions and configurations during a session. These operating conditions and configurations are the result of negotiation, either by one of the communicants, such as a user, a machine that is
35 associated with the user, or by a neutral party. The self-descriptions can be used to negotiate a communication

paradigm with an information system. The negotiations may be simple, *e.g.*, matching attributes, or complex, *e.g.*, matching an offer to a range (*e.g.*, similarity), weighted decision models and counter-offers.

The self-descriptions, or "profiles", include: information about the user's working context; about the task that is to be performed; what kinds of constraints are placed on how the task is accomplished, *e.g.*, by the working context or by limitations of the user; what solutions the user prefers to apply to ameliorate the constraints; platform and communications capacities; device capabilities and limitations. However, not all sessions will use or even have this information, depending on what the interacting components are and what is to be done. For example, in coupling a headtracking device to a computer through a wireless connection, a profile might only state that for all contexts, a selected user prefers the head tracker as a pointing device to be used for control and data entry, and that visual presentation is preferred in all contexts. Note that no mention is made of a disability for the user. In fact, this user might not be disabled at all, but merely does not have hands free to operate a mouse and keyboard. For privacy considerations, the characterization of a user's disability need not be present in a profile.

The foregoing and still other benefits of the adaptation system 100 include: 1) communications processes that permit conveying user preference and capability information to an adaptation service; 2) software for use as an adaptation service to generate transformations of user interface and content of applications and any other information necessary for operational configuration and setup of the user's system on the basis of a user's preferences, capabilities and situation; 3) communications for conveying said generated transformation and operational information to the user's system or to a another system acting as proxy for the user's system; 4) a protected storage system for holding users' personal data related to capabilities and preferences; 5) an extraction mechanism which abstracts a user's personal data to protect privacy, for producing a statement of preferences and capabilities in a current situation to be conveyed to an adaptation service; 6) a mechanism for learning about a user's preferences and capabilities and using this information for optimizing adaptation processes. The invention employs several aspects of prior art in novel ways and introduces new functionality. In this regard, the invention provides an end-to-end process beginning with gathering user information, through generating transformations based upon that information, conveying the instructions for changes to the user system for implementing the changes in the user's system.

More complicated profiles can occur because the user is affected by noise or heat, for example, or because a tactical military situation prevents high light levels on a screen. A user might also find after a while that he or she is becoming fatigued or that the cognitive load in the presentation is too high. Again, these could be related to disabilities or not at all. The profile is able to handle various levels of complexity.

Explicit negotiation, rather than simple attribute/capability matching may be used in some systems. For example, in ATMs and public kiosks, there may be many hundreds of different people who use the device per day. Each of these persons would have their own particular way (plus local environmental conditions) of interacting with the system, so that many different profiles might need to be negotiated. Contrast this with a personal computer (PC) which may have only one user, with an essentially non-varying environment. Here the negotiation may need to be done only once, when the system is first brought up. However, if the PC is on a network, then a negotiation may be

necessary with each different system on the net with which the person wants to interact. Now consider a home network in which each person in a home has their own preferred way of interacting with the appliances: children different than parents, different than grandparents, different than family member with disability. Negotiation may be necessary as each new appliance (or new person) is added, but after that operation can use profiles extracted immediately from a local repository. However, if a new family should move into the house, negotiation might have to begin anew, even if the new family imports its profiles from a previous residence.

Sometimes a disability is irrelevant to a condition in a context. For example, a blind person who can read Braille is generally able to read whether it is dark or light, quiet or noisy in the surrounding environment. On the other hand, a disability might be irrelevant in one interaction function but crucial in another. A person who has only a motor nerve difficulty (e.g., quadriplegic) might not be constrained by this disability in visual or aural presentations, but might be considerably constrained when control has to be actuated through the visual presentation—for example, navigating and selecting hypertext links in small font on the browser screen. A disability, by its nature, may also cause temporary impairment of an ability that is not otherwise constrained. For example, a person in a wheelchair, because of a bad design of the area around an ATM, might not be able to get close enough to the ATM to see the display or to operate the controls, even though the person is not blind and has full use of hands. Consequently, any impairment should be carefully correlated to task and context for appropriate accommodation.

When interfaces are to be implemented with multi-modal interactions, collaboration among the various modal accessing elements can be performed. An example of this is a military command post where speech, visual, audio and gestural controls are coordinated. Each officer will have preferences based upon rank, experience, specialty and mission assignment, in addition to personal effects. There adaptation system can handle collaborations among two or more users. If the adaptation system is not powerful enough to execute the transformations by itself, a proxy can be used to handle this execution for the adaptation system. Consequently, configuration information may accompany the instructions for interface transformations. Similarly, additional communications or protocols may be used to accomplish the execution. For example, if a proxy is used, an identifier and locator may be used to provide access to it, and possibly another protocol invoked between the proxy and the user system. Consequently, some setup information may also need to accompany the transformation instructions.

Preferences need not be restricted to apparent interface changes. For example, a user is driving in a car and receives a long email, or an email with a voluminous attachment, perhaps an image. The user has voice/audio interaction in place but because of the size of the email, and (perhaps for his own safety and that of other drivers) he has previously designated that such mail be redirected to his home or office system where it can be read visually. This is a transaction preference. It engenders, in a sense, an alternative interface action, because it delays the presentation of the content and chooses an alternative outside the current user system. Moreover, there is configuration and setup information generated as well, but the current user system does not receive it. The home or office system acts as a proxy for the current use system.

The present system allows for the generation of self-descriptions that define user preferences and the capabilities of components interacting on some task and their operating environments and conditions. The adaptation system 162 enables negotiations based on those descriptions and conditions, and the construction of configurations and interface transformations according to the negotiations for accomplishing the intended task.

5 While the above detailed description has shown, described, and pointed out novel features of the invention as applied to various embodiments, it will be understood that various omissions, substitutions, and changes in the form and details of the device or process illustrated may be made by those skilled in the art without departing from the spirit of the invention. The scope of the invention is 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
10 within their scope:

WHAT IS CLAIMED IS:

1. A method of adapting a predefined interface, the method comprising:
identifying a preference object that includes at least one user characteristic from a set of user characteristics, wherein the set of user characteristics comprises at least one of situational information, environmental information, behavior information, and context information;
5 transmitting a first capability object from a selected electronic device to an adaptation engine (162), wherein the first capability object defines one or more attributes regarding the electronic device;
transmitting a second capability object to the adaptation engine (162), wherein the second
10 capability object defines one or more attributes of an information source (166) that is accessed by the electronic device; and
generating an adaptation object, based at least in part upon the at least one user characteristic, the contents of the transmitted first capability object, and the contents of the transmitted second capability object, wherein the adaptation object defines at least one adaptation rule for adapting human factor
15 information that is transmitted by the electronic device or the information source (166).
2. The method of Claim 1, additionally comprising applying the adaptation rule thereby adapting the human factor information that is transmitted by the electronic device.
3. The method of Claim 1, additionally comprising transmitting the adaptation object to the electronic
20 device, wherein the electronic device uses the adaptation object to adapt an interface of the electronic device.
4. The method of Claim 1, additionally comprising transmitting the adaptation object to a proxy (136)
for the electronic device.
25
5. The method of Claim 1, wherein the adaptation object is transmitted to the information source (166), and wherein the information source (166) uses the adaptation object to adapt the communication between the information source (166) and the accessed electronic device.
6. The method of Claim 1, wherein the selected electronic device includes an application program, and
30 wherein the second capability object is associated with the application program.
7. An adaptation engine (162), comprising:
a communication interface (204) configured to receive a plurality of preference objects and a
35 plurality of capability objects, wherein the preference objects define at least one preference of a user, and wherein the capability objects each define the capabilities of a plurality of electronic devices; and

at least one adapter service unit (228) that generates, based at least in part upon the contents of the preference objects and the contents of the capability objects, a set of rules for modifying one or more aspects of an interface between the electronic devices and the user.

5 8. The adaptation engine (162) of Claim 7, additionally comprising a consolidator (232) that eliminates any conflicts in the set of rules by applying a set of consolidator rules.

 9. The adaptation engine (162) of Claim 7, wherein the communication interface 204 automatically receives the preference objects and the capability objects in response to a user accessing one of the electronics
10 devices.

 10. An adaptation object residing in a memory of an electronic device, the adaptation object comprising:

 a plurality of adaptation instructions including at least one rule that specifies a change to an
15 interface between a user and an electronic device;

 wherein the adaptation object is generated in real time based in response to the user requesting to use a computing device to retrieve information from an information source (166); and

 wherein the contents of the adaptation object are based at least in part upon the content of a capability object that is associated with the electronic device, the content of a capability object that is
20 associated with the information source (166), and the content of a preference object that defines one or more preferences of the user.

 11. A system for adapting a predefined interface, the system comprising:

 means for selecting at least one user characteristic from a set of user characteristics, wherein the
25 set of user characteristics comprises situational information, environmental information, behavior information, and context information;

 means for transmitting a first capability object from a selected electronic device to an adaptation engine (162), wherein the first capability object defines one or more attributes regarding the electronic device;

30 means for transmitting a second capability object to the adaptation engine (162), wherein the second capability object defines one or more attributes of an information source (166) that is accessed by the electronic device; and

 means for generating, based at least in part upon the at least user characteristic, the transmitted first capability object, and the transmitted second capability object, an adaptation object, wherein the

adaptation object defines at least one adaptation rule for adapting human factor information that is transmitted by the electronic device or the information source (166).

12. The system of Claim 11, additionally comprising means for applying the adaptation rule thereby
5 adapting the human factor information that is transmitted by electronic device.

13. The system of Claim 11, additionally comprising means for transmitting the adaptation object to
the electronic device, wherein the information source (166) uses the adaptation object to adapt the communication
between the information source (166) and the electronic device.

10 14. The system of Claim 11, additionally comprising means for transmitting the adaptation object to a
proxy (136) for the accessed electronic device.

15 15. The system of Claim 11, wherein the adaptation object is transmitted to the information source
(166), and wherein the information source (166) uses the adaptation object to adapt the communication between the
information source (166) and the accessed electronic device.

16. A method of generating an adaptation object, the method comprising:
identifying one or more user preferences;
20 identifying the capabilities of a first device;
identifying the capabilities of a second device; and
determining whether to modify an interface of the first device, wherein the determining is based at
least in part upon the identified user preferences, the identified capabilities of the first device, and the
identified capabilities of the second device.

25 17. The method of Claim 16, additionally comprising generating an adaptation object, wherein the
generating is based at least in part upon the identified user preferences, the identified capabilities of the first device,
and the identified capabilities of the second device.

30 18. The method of Claim 17, additionally comprising transmitting the adaptation object to the first
electronic device, wherein the first electronic device uses the adaptation object to modify the interface of the first
device.

35 19. The method of Claim 17, additionally comprising transmitting the adaptation object to the second
electronic device, wherein the second electronic device uses the adaptation object to modify the interface of the first
device.

20. The method of Claim 17, additionally comprising transmitting the adaptation object to a proxy (136), wherein the proxy (136) uses the adaptation object to modify the interface of the first device.

5 21. The method of Claim 16, wherein the selected electronic device includes an application program, and wherein the second capability object is associated with the application program.

22. An adaptation system comprising:
a user input description module (124) for receiving preference information that defines preferentially one or more types of presentation that are preferred by the user;
10 a preference assembly module (154) for building a preference object that is based at least in part upon the preference information;
an electronic device that is accessible by the user;
an electronic device capability object that is associated with the electronic device, wherein the electronic device capability objects defines the capabilities that are associated with the electronic device;
15 an information source (166) that is accessible by the electronic device;
an information source capability object that is associated with the information source (166), wherein the information source capability object defines the capabilities that are associated with the information source (166); and
20 an adaptation engine (162) for generating, based at least in part upon the preference object, the electronic device capability object, and the information source capability object, an adaptation object, wherein the adaptation object defines at least one adaptation rule for adapting human factor information that is transmitted by the electronic device or the information source (166).

23. A program storage device storing instructions that when executed perform the acts comprising:
25 identifying one or more user preferences;
identifying the capabilities of a first device;
identifying the capabilities of a second device; and
determining whether to modify an interface of the first device, wherein the determining is based at least in part upon the identified user preferences, the identified capabilities of the first device, and the
30 identified capabilities of the second device.

24. The program storage device of Claim 23, additionally storing instructions that when executed perform the act comprising generating an adaptation object, wherein the generating is based at least in part upon the identified user preferences, the identified capabilities of the first device, and the identified capabilities of the second
35 device.

25. The program storage device of Claim 23, additionally storing instructions that when executed perform the act comprising transmitting the adaptation object to the first electronic device, wherein the first electronic device uses the adaptation object to modify the interface of the first device.

5 26. The program storage device of Claim 23, additionally storing instructions that when executed perform the act comprising transmitting the adaptation object to the second electronic device, wherein the second electronic device uses the adaptation object to modify the interface of the first device.

10 27. The program storage device of Claim 23, additionally storing instructions that when executed perform the act comprising transmitting the adaptation object to a proxy (136), wherein the proxy (136) uses the adaptation object to modify the interface of the first device.

15 28. A preference object residing in a memory of an electronic device, the preference object comprising:
at least two preference functions, wherein each of the at least two preference functions have an associated type, wherein the type is one of: entry, control, presentation, and authorization;
at least one preference rating for each of the preference functions, wherein the preference rating preferentially orders the at least two preference functions.

20 29. The preference object of Claim 28, wherein the preference object is adapted for use in an adaptation engine (162) that generates an adaptation object based at least in part upon the contents of the preference object.

30 30. The preference object of Claim 28, wherein the preference function comprises a type descriptor identifying a type of the preference function.

25 31. The preference object of Claim 28, wherein the preference function comprises a name descriptor identifying a unique name that is associated with the preference function.

32. The preference object of Claim 28, wherein the preference function comprises a mode descriptor identifying a mode that is associated with the preference function.

30

33. A method of generating a preference object in an electronic device, the method comprising:

receiving a user description object that defines user characteristic information, wherein the user characteristic information is selected from the group comprising: situational information, environmental information, behavior information, and context information;

35

storing the user description object in a database;

determining the operating contextual conditions of the user; and

generating, in response to a user accessing an electronic device, a preference object that defines one or more user preferences, wherein the content of the preference object is at least based in part upon the content of the user description object and the determined operating contextual conditions.

5 34. The method of Claim 33, additionally comprising storing, in the preference object, an object identifier of an electronic device that generated the preference object

 35. The method of Claim 33, additionally comprising storing, in the preference object, an adaptation session descriptor that identifies a session, wherein the session is associated with the user that is accessing the electronic
10 device.

 36. A program storage device storing instructions that when executed performs the acts comprising:
 receiving a user description object that defines user characteristic information, wherein the user
 characteristic information is selected from the group comprising: situational information, environmental
15 information, behavior information, and context information;
 storing the user description in a database;
 determining the operating contextual conditions of the user; and
 generating, in response to a user accessing an electronic device, a preference object that defines
 one or more user preferences, wherein the content of the preference object is at least based in part upon the
20 content of the user description object and the determined operating contextual conditions.

 37. The program storage device of Claim 36, additionally comprising instructions that when executed
perform the act of storing, in the preference object, an object identifier of an electronic device that generated the
preference object

25

 38. The program storage device of Claim 36, additionally comprising instructions that when executed
perform the act storing, in the preference object, an adaptation session descriptor that identifies a session, wherein
the session is associated with the user accessing the electronic device.

30 39. A system for generating a preference object in an electronic device, the method comprising:
 means for receiving a user description object that defines user characteristic information, wherein
 the user characteristic information is selected from the group comprising: situational information,
 environmental information, behavior information, and context information;
 means for storing the user description in a database;
35 means for determining the conditions of the user; and

means for generating, in response to a user accessing an electronic device, a preference object that defines one or more user preferences, wherein the content of the preference object is at least based in part upon the content of the user description object and the determined conditions.

5 40. The system of Claim 39, additionally comprising means for storing, in the preference object, an object identifier of an electronic device that generated the preference object

 41. The system of Claim 39, additionally comprising means for storing, in the preference object, an adaptation session descriptor that identifies a session, wherein the session is associated with the user accessing the
10 electronic device.

 42. A capability object residing in a memory of an electronic device, the capability object comprising:
 a descriptor which identifies an electronic device or software application that is associated with the capability object and which identifies whether the electronic device or software application is directly
15 accessible by the user or, alternatively, an information source (166) that is accessible by the electronic device or software application; and
 at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object as being either (1) an electronic device or software application or (2) an
 information source (166).

20 43. The capability object of Claim 42, additionally comprising an adaptation object listener identifier that identifies an adaptation device that receives any adaptation object that is generated for use with the electronic device.

 44. The capability object of Claim 42, wherein the capability comprises:
25 one or more characteristics of the capability; and
 an identifier of an entity that is designated to receive any adaptation object that is generated based at least in part upon the contents of the capability object.

 45. The capability object of Claim 42, wherein the type descriptor is one of: entry, control, presentation,
30 authorization, application, operating system, processor system, and communications.

 46. A method of making a capability object, the method comprising:
 storing, in a memory, a descriptor which identifies an electronic device or a software application that is associated with the capability object and which identifies whether the electronic device or software
35 application is associated is accessible by the user or, alternatively, an information source (166) that is accessible by the electronic device or software application; and

storing, in the memory, at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object.

5 47. The method of Claim 46, additionally comprising storing, in the memory, one or more characteristics of the capabilities of the capability object.

10 48. The method of Claim 47, additionally comprising storing, in the memory, an identifier of an entity that is designated to receive any adaptation object that is generated based at least in part upon the contents of the capability object.

15 49. A system for making a capability object, the system comprising:

means for storing, in a memory, a descriptor which identifies an electronic device or a software application that is associated with the capability object and which identifies whether the electronic device or software application is accessible by the user, or, alternatively, an information source (166) that is accessible by the electronic device; and

means for storing, in the memory, at least one capability function, the capability function comprising a type descriptor that defines the type of the capability object.

20 50. The system of Claim 49, additionally comprising means for storing, in the memory, one or more characteristics of the capabilities of the capability object.

25 51. The system of Claim 49, additionally comprising storing, in the memory, an identifier of an entity that is designated to receive any adaptation object, that is generated based at least in part upon the contents of the capability object.

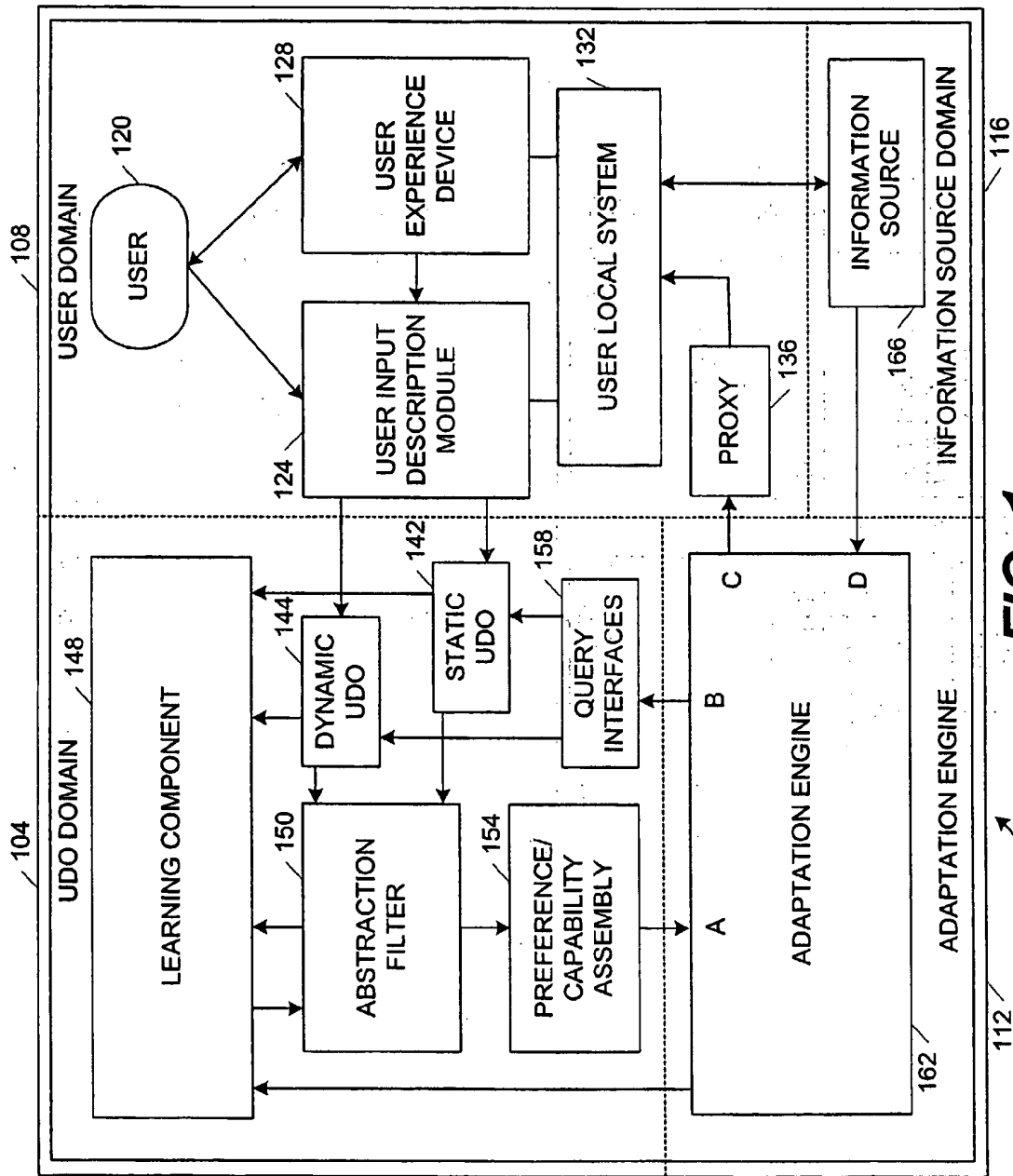


FIG. 1

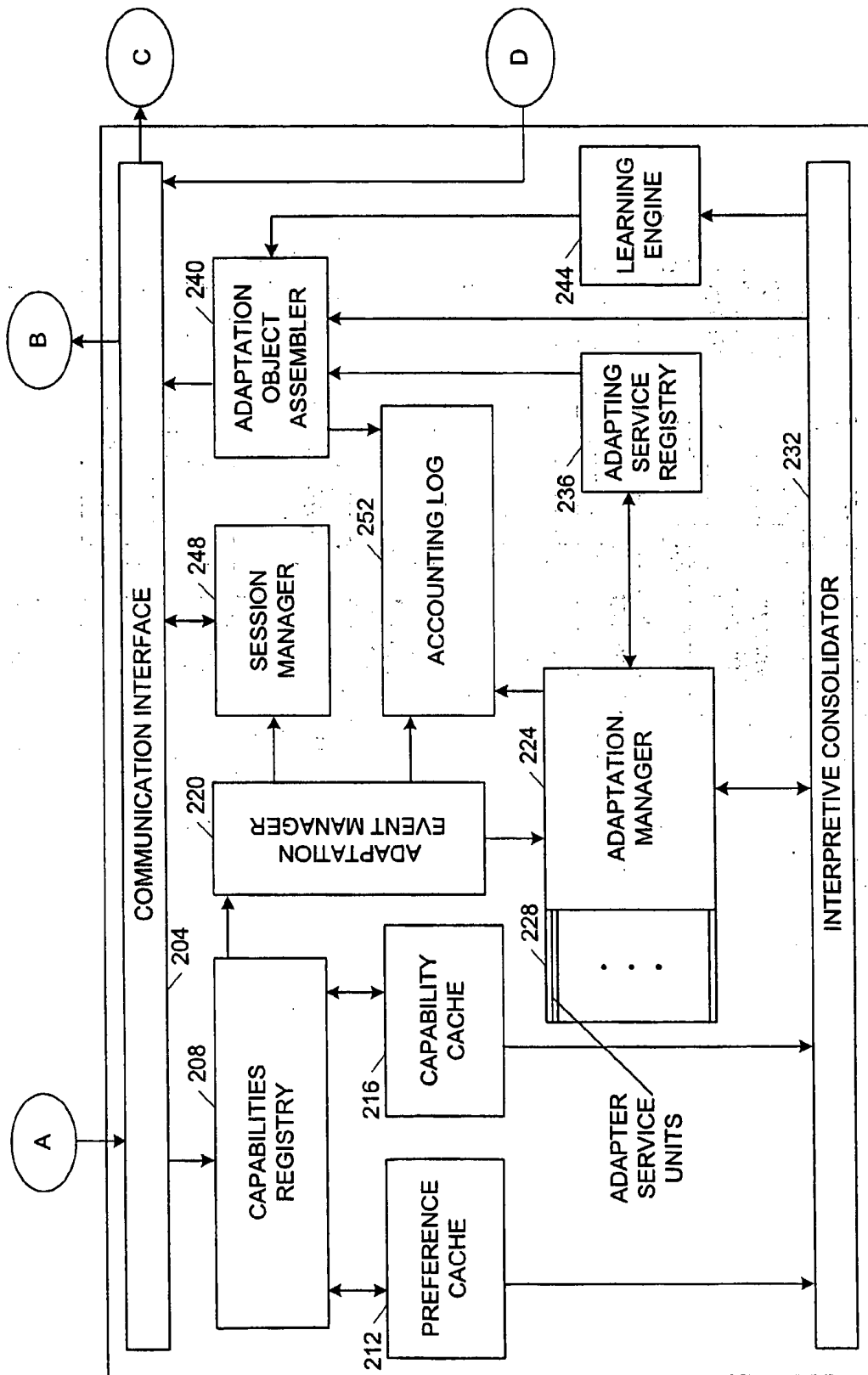


FIG. 2

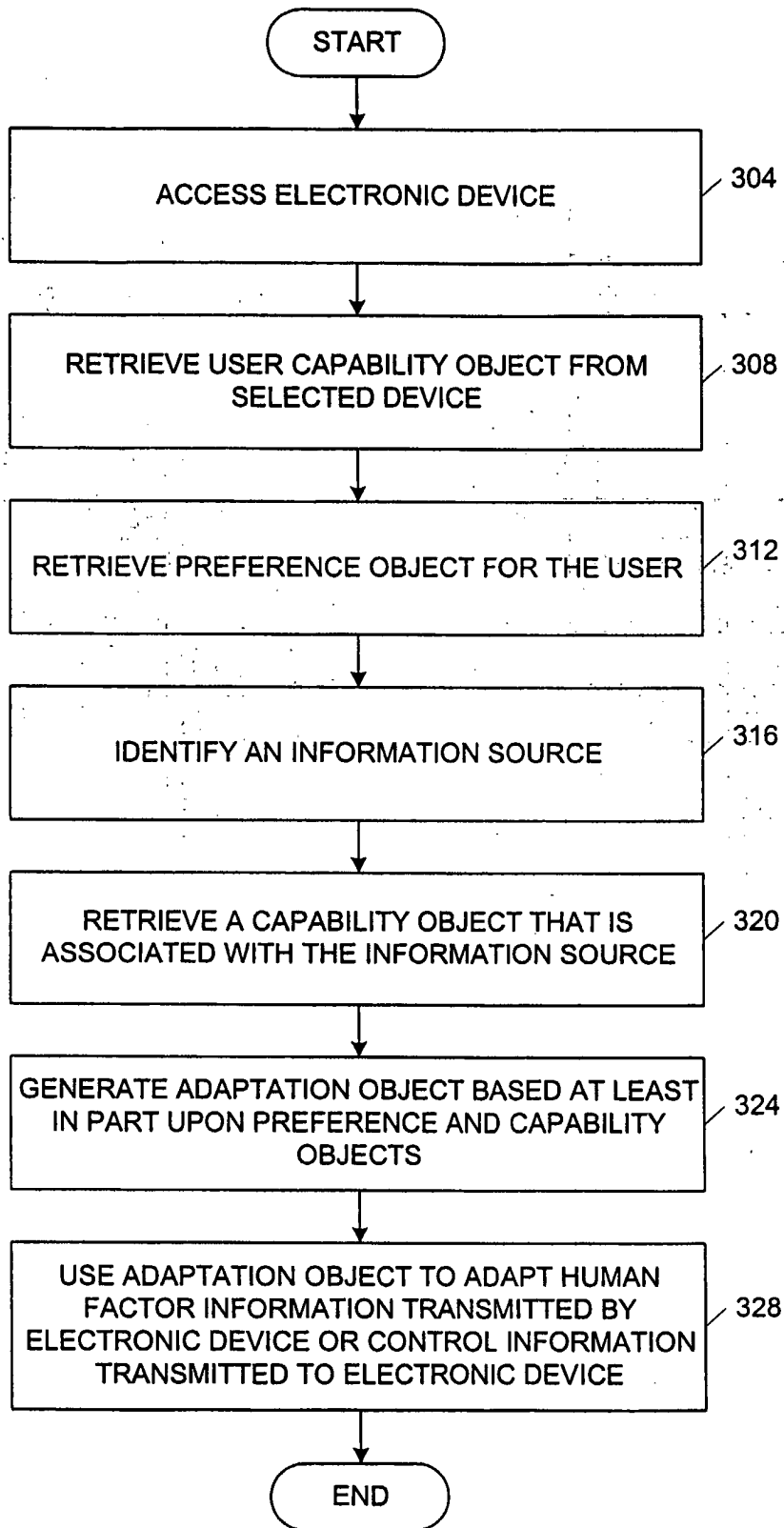


FIG. 3

4/9

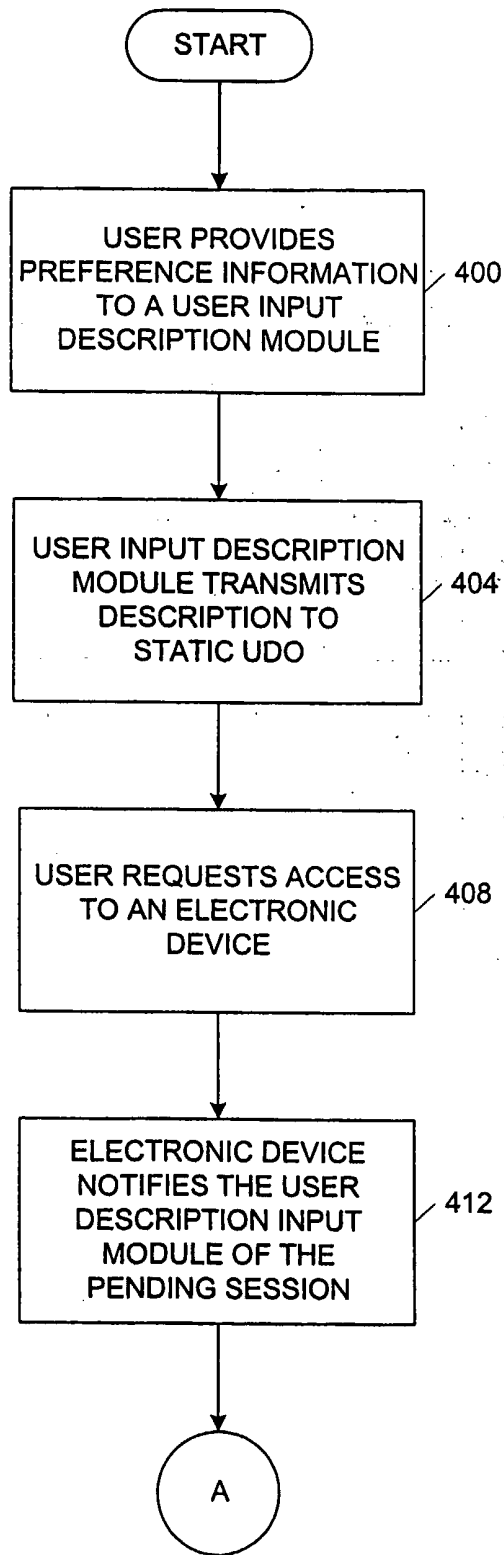


FIG. 4A

5/9

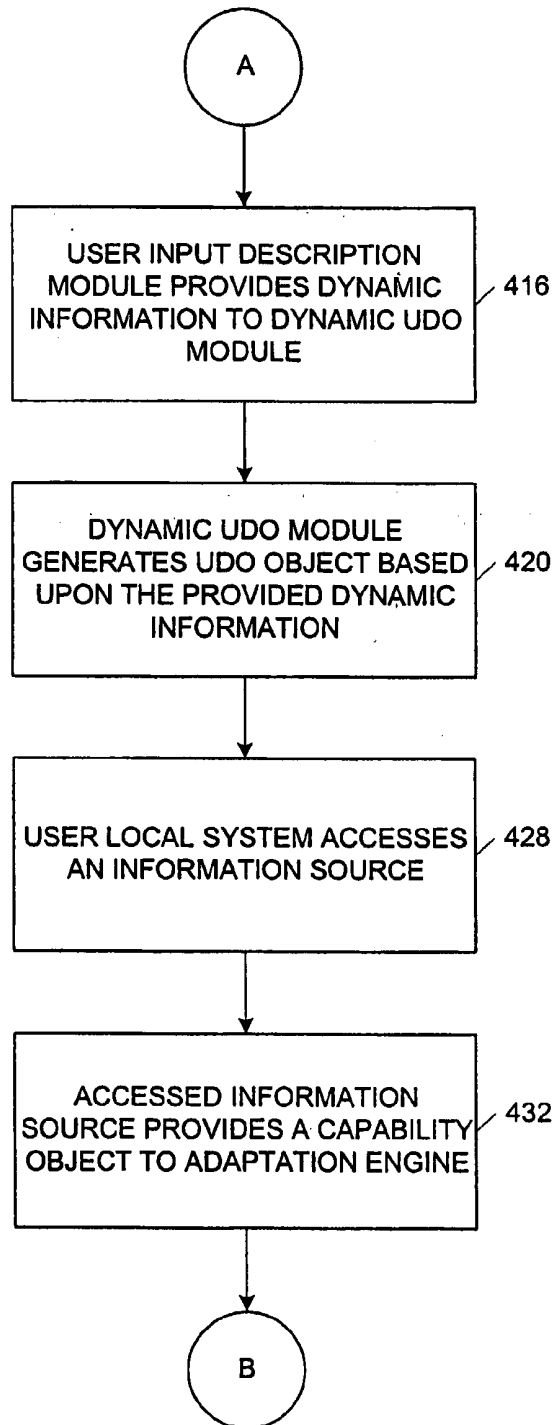


FIG. 4B

6/9

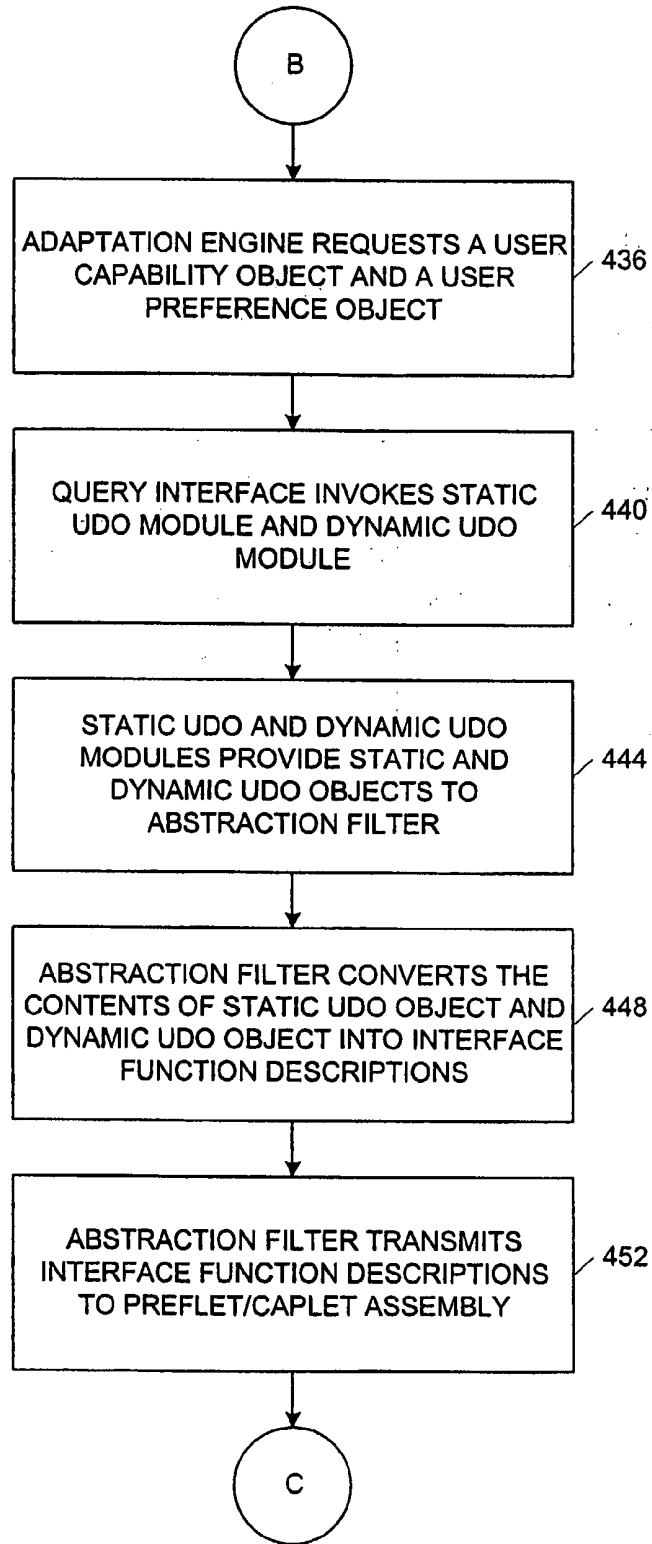


FIG. 4C

7/9

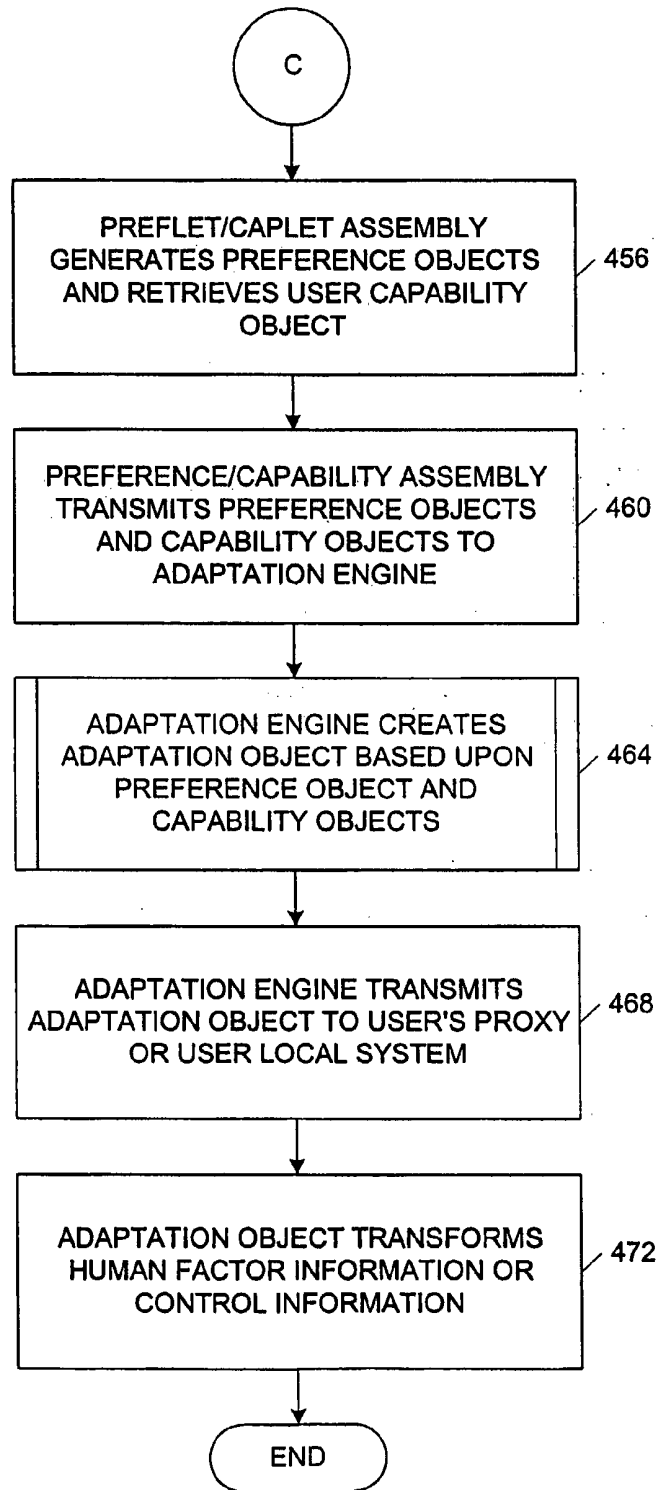


FIG. 4D

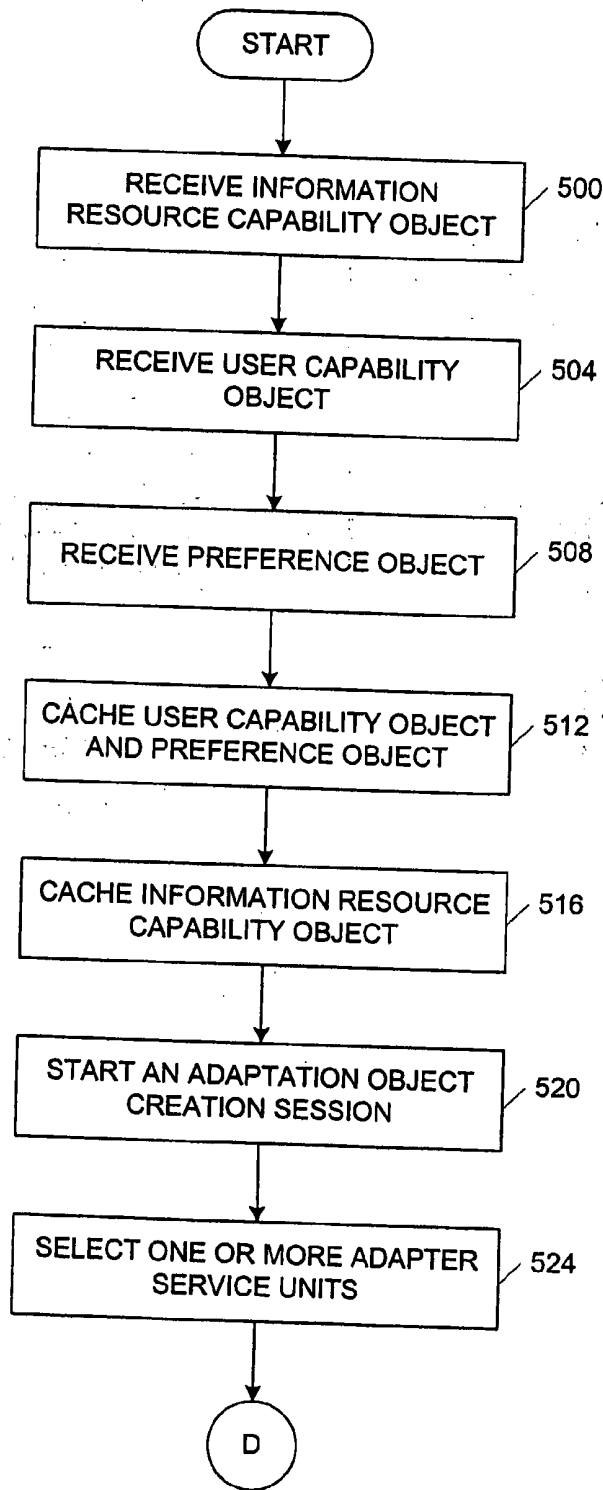


FIG. 5A

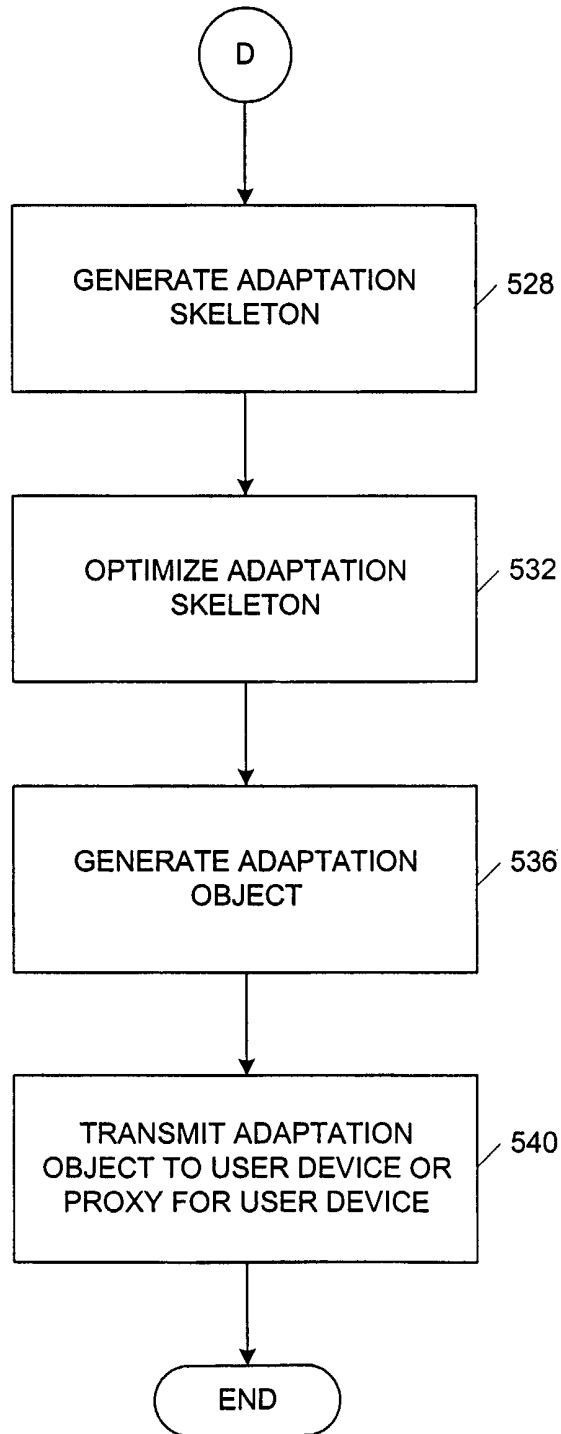


FIG. 5B