

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
6 July 2006 (06.07.2006)

PCT

(10) International Publication Number
WO 2006/070074 A1

(51) International Patent Classification:
G06F 3/01 (2006.01)

(21) International Application Number:
PCT/FI2005/050487

(22) International Filing Date:
27 December 2005 (27.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/026,447 30 December 2004 (30.12.2004) US

(71) Applicant (for all designated States except US): **NOKIA CORPORATION** [FI/FI]; Keilalahdentie 4, FI-02150 Espoo (FI).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **SALMINEN, Henri** [FI/FI]; Kevätniitynkuja 5, FI-36100 Ruutana (FI).

(74) Agent: **KOLSTER OY AB**; Iso Roobertinkatu 23, P.o.box 148, FI-00121 Helsinki (FI).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:

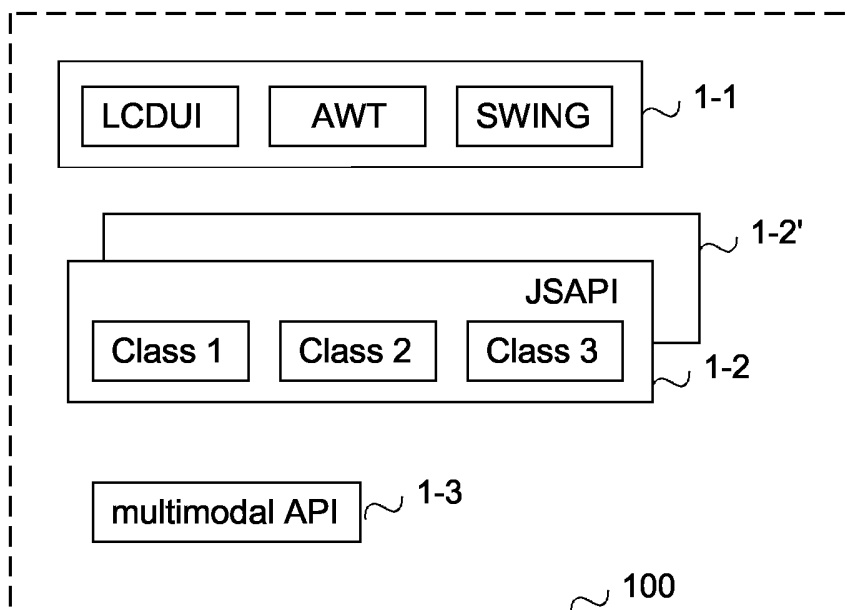
— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

— with international search report

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.

(54) Title: MULTIMODAL INTERACTION



(57) Abstract: In order to enable an application to be provided with mul- timodal inputs, a multimodal application interface (API), which contains at least one rule for providing multimodal interaction is provided.

WO 2006/070074 A1

MULTIMODAL INTERACTION

FIELD OF THE INVENTION

[0001] The present invention relates to multimodal interaction.

BACKGROUND OF THE INVENTION

[0002] Output and input methods of user interfaces in applications, especially in browsing applications, are evolving from standalone input/output interaction methods to user interfaces allowing multiple modes of interaction, such as means for providing input using voice or a keyboard and output by viewing and listening. To enable this, mark-up languages are being developed. For the time being, solutions with different modalities being used to access a service at different times are known and multimodal service architectures with co-operating voice and graphical browsers are evolving.

[0003] Although multimodal browsing is evolving, utilizing multiple input modalities (channels) in software applications has not been brought into focus. Solutions developed for mark-up languages cannot be used with software applications as such, since a mark-up language is used for describing the structure of structured data, based on the use of specified tags, whereas a software application actually processes the data (which may be in a mark-up language), and therefore the requirements are different. In a software application capable of receiving inputs from two or more separate modalities, synchronization between different modalities is needed. For example, in order to perform one uniform controlling action of a software application, a user may have both to speak and point an item within a timeframe. Since the accuracy and lag between different modalities vary, timing might become crucial. This is a problem not faced at a mark-up language level with multimodal browsing since the internal implementation of a browser takes care of the timing, i.e. each browser interprets a multimodal input in its own way.

[0004] One solution is to implement multimodal interaction of a software application in a proprietary way. A problem with this solution is that every software application, which utilizes multimodal interaction, needs to be implemented with a separate logic for the multimodal interaction. For example, accuracy issues should be taken into account by confirmation dialogs. Thus, quite complex tasks are left for an application developer to solve.

BRIEF DESCRIPTION OF THE INVENTION

[0005] An object of the present invention is to provide a method and an apparatus for implementing the method so as to overcome the above problem. The object of the invention is achieved by a method, an electronic device, an application development system, a module and a computer program product that are characterized by what is stated in the independent claims. Preferred embodiments of the invention are disclosed in the dependent claims.

[0006] The invention is based on the idea of realizing the need for a mechanism supporting a multimodal input and the above problem and providing a high-level structure called a multimodal application programming interface (API) containing one or more rules for multimodal interaction, the rule or rules manipulating inputs according to one or more rules. A rule may concern one modality or it may be a common rule concerning at least two different modalities.

[0007] An advantage of the above aspect of the invention is that it enables an application developer to design applications with multimodal control user interfaces in the same way as graphic user interfaces.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] In the following, the invention will be described in greater detail by means of exemplary embodiments with reference to the accompanying drawings, in which

Figure 1 illustrates an example of an application development system according to an exemplary embodiment of the invention;

Figure 2 is a block diagram of a multimodal API according to a first exemplary embodiment of the invention;

Figures 3A and 3B show a pseudocode of a multimodal API according to the first exemplary embodiment of the invention;

Figure 4 is a flow chart illustrating a simplified example of application creation with the multimodal API according to the first exemplary embodiment of the invention;

Figure 5 shows a pseudocode indicating how the multimodal API of Figures 3A and 3B can be used;

Figures 6 and 7 are flow charts illustrating different implementations of the multimodal API;

Figure 8 is a block diagram of a multimodal API according to a sec-

ond exemplary embodiment of the invention;

Figure 9 is a flow chart illustrating a simplified example of application creation with the multimodal API according to the second exemplary embodiment of the invention;

Figure 10 shows a pseudocode indicating how the multimodal API according to the second exemplary embodiment of the invention can be used;

Figure 11 is a flow chart illustrating the use of the multimodal API according to the second exemplary embodiment of the invention;

Figure 12 is a simplified block diagram of a module; and

Figure 13 is a simplified block diagram of a device.

DETAILED DESCRIPTION OF SOME EMBODIMENTS

[0009] The present invention is applicable to any application development system supporting multimodal controlling, and to any software application/module developed by such a system and to any apparatus/device utilizing multimodal controlling. Modality, as used herein, refers to an input or an output channel for controlling a device and/or a software application. Non-restricting examples of different channels include a conventional mouse, a keyboard, a stylus, speech recognition, gesture recognition and haptics recognition (haptics is interaction by touch), input from an in-car computer, a distance meter, a navigation system, a cruise control, a thermometer, a hygrometer, a rain detector, a weighing appliance, a timer, machine vision, etc.

[0010] In the following, the present invention will be described using a system relying on a Java programming language environment as an example of a system environment whereto the present invention may be applied, without restricting the invention thereto; the invention is programming language independent.

[0011] Figure 1 illustrates architecture of an application development system 100 according to an embodiment of the invention. The exemplary application development system comprises graphic user interface (GUI) frameworks 1-1, different modality APIs 1-2, 1-2' and a multimodal API 1-3. The existing GUI frameworks and future GUI frameworks may be utilized with the invention as they are (and will be), the invention does not require any changes to them, or set any requirements for them either. The same also applies to different modality APIs.

[0012] A number of GUI frameworks 1-1 exist for Java, such as those illustrated in Figure 1, Swing, AWT (Abstract Window Toolkit) and LCDUI (liquid crystal display user interface for Java 2 Micro Edition (J2ME), i.e. for wireless Java applications), for example. Each GUI framework contains classes (not illustrated in Figure 1). It should be noted that the GUI frameworks here are just examples, any other frameworks may be used instead of or with a GUI framework.

[0013] In the example shown in Figure 1, only one of the modality APIs is shown in detail, the modality API being Java Speech API JSAPI 1-2 containing different classes.

[0014] The multimodal API 1-3 provides an integration tool for different modalities according to the invention and different embodiments of the multimodal API 1-3 will be described in more detail below. The multimodal API 1-3 can be used in several applications in which multimodal inputs are possible, including but not limited to applications in mobile devices, vehicles, airplanes, home movie equipment, automotive appliances, domestic appliances, production control systems, quality control systems, etc.

[0015] A first exemplary embodiment of the invention utilizes aspect-oriented programming. Aspect-oriented programming merges two or more objects into formation of the same feature. Aspects are abstractions of the same kind as classes in object-oriented programming, but aspects are intended for cross-object concerns. (A concern is a particular goal, concept or area of interest and a crosscutting concern tends to affect multiple implementation modules.) Thus, aspect-oriented programming is a way of modularizing crosscutting concerns much like object-oriented programming is a way of modularizing common concerns. A paradigm of aspect-oriented programming is described in US patent 6 467 086, and examples of applications utilizing aspect oriented programming are described in US patent 6 539 390 and US patent application 20030149959. The contents of said patents and patent application are incorporated herein by reference. Information on the aspect-oriented programming can also be found via the Internet pages <http://www.javaworld.com/javaworld/jw-01-2002/jw-0118-aspect.html> and <http://eclipse.org/aspectj/>, for example.

[0016] Figure 2 illustrates the multimodal API according to the first exemplary embodiment of the invention in which the multimodal API is provided by one or more multimodal aspects, later called aspects. Depending on

the implementation, the multimodal API comprises one or more aspects. An aspect represents integration of modalities into one interaction. Each aspect contains one or more rules to perform a multimodal interaction. For example, aspect 1 may be an aspect for integrating speech with gestures, aspect 2 may be an aspect for integrating speech with text given via a graphical user interface, and aspect N an aspect for integrating speech with gestures and with text. There may be an aspect integrating outputs of other aspects, i.e. aspects may be chained. Yet another possibility is that there is only one universal aspect integrating all possible multimodal inputs. Aspects may be implemented with a Java extension called AspectJ. An example of an aspect, a pseudocode for an aspect integrating speech and mouse input, i.e. two different ways to select an option from a text box, is shown in Figures 3A and 3B which form a single logical drawing. The pseudocode is loosely based on CLDC (Connected Limited Device Configuration) 1.0 and MIDP (Mobile Information Device Profile) 1.0, JSAPI 2.0 and AspectJ. A J2ME environment is formed by first creating a configuration containing basics, types, data structures, etc., and creating then, on the configuration, a profile containing higher-level features, such as LCDUI. As can be seen, an aspect 300 contains the actual integration and decides what is integrated and what is not, thus guaranteeing that the application program is controlled by synchronized and accurate modalities. By using this aspect, an application developer need not to worry about these details any more. As can be seen in the example illustrated in Figure 3, the aspect contains one or more rules 305 for different modalities. Different modalities to be integrated are defined in section 301, utility functions used with them are defined in 302, sections 303 and 304 define modality-specifically how recognition is performed.

[0017] Figure 4 is a flow chart illustrating a simplified example of how an application developer can create an application utilizing the multimodal API according to the first exemplary embodiment. The application here is a multimodal user interface, such as a mobile information device applet (MIDlet). First, the application developer selects one or more suitable classes from a GUI framework (step 401), and one or more modalities APIs (step 402). The application developer then selects, in step 403, one or more suitable classes for each selected GUI framework and for each selected modality API. The application developer may have selected a text box implemented by LCDUI and a speech recognizer implemented by JSAPI. The application developer then se-

lects, in step 404, a suitable aspect or suitable aspects for multimodal interaction and the application is ready. However, the application developer may configure the selected aspect(s) if needed. Thus, by selecting an aspect, a rule is selected but by configuring the aspect, the selected rule may be fine-tuned when necessary. In one embodiment of the invention, the rules of aspect may be dynamic, i.e. rules are modified according to the input they receive. This input may comprise, but is not limited to, the input from the modality and/or other information, such as delay in speech recognition, reliability of speech recognition result, error messages inputted by the user or some other computer program module, or time interval of receiving input from two modalities, for example.

[0018] An example of how the application developer may use the aspect shown in Figures 3A and 3B is illustrated by the pseudocode in Figure 5. Section 501 illustrates the outcome of the above-described steps 401 to 403, section 502 illustrates the outcome of the step 404 described above, and section 503 defines a tool to be used when the selections of different modalities (speech and text) will be mapped to each other. Section 504 gives some explanatory information commenting how the aspect functions, i.e. how the application receives interactions. The commented functionality is within the aspect. As can be seen, the aspect provides a guideline for multimodal interaction which can then be tuned by configuring the selected aspect.

[0019] Figure 6 is an exemplary flow chart illustrating with a simplified example a first implementation of the multimodal API according to the first exemplary embodiment. For the sake of clarity, it is assumed that the application may receive inputs from two different modalities. This first implementation is referred as multimodal API utilizing aspect-oriented programming.

[0020] Figure 6 starts when the multimodal API receives an input from a modality API 1 in step 601. In response to the received input, the multimodal API checks, in step 602, whether the input relates to multimodal interaction. If it does not relate to a multimodal event, the input is sent, in step 603, to the application. If the input relates to a multimodal event, the input is forwarded, in step 604, to another modality API according to associated rule in the modality API. The other modality API then recognizes that the input was received from the modality API and sends this received input as its own input to the application in request. In this exemplary embodiment, the multimodal API acts as an aspect, which handles the crosscutting concerns of different

modalities. It provides a mechanism that only one input to a requesting application is obtained. The aspect handles and forwards the data it receives from the modalities according to rules.

[0021] Figure 7 is a flow chart illustrating a second implementation of the multimodal API according to the first exemplary embodiment with a simplified example. Also here it is assumed, for the sake of clarity, that the application may receive inputs from two different modalities. This first implementation is referred to as multimodal integrator.

[0022] Figure 7 starts when the multimodal API receives an input from a modality API 1 in step 701. In response to the received input, the multimodal API checks, in step 702, whether the input relates to a multimodal event. If it relates to a multimodal event, the multimodal API waits, in step 703, a preset time for an input from the other modality API, modality API 2. The waiting time, i.e. a preset time limit, may be set when the multimodal API is being created. In another exemplary embodiment of the invention, the multimodal API may take into account also other data it receives from modalities, lag in speech recognition, trustworthiness of speech recognition result, error messages received from the user or from other API's or computer program product, for example. The rules, which are used to integrate input from the modalities, may also be dynamic, i.e. the rules are modified according to information the multimodal API receives. If the other input is received (step 504) within the time limit, the multimodal API integrates, in step 705, the inputs together into one integrated input, and sends the input to the application in step 706. One example of integration is given: Let us assume that coffee is selected via a graphical user interface GUI, and after a few seconds, a selection "tea" is received via speech recognition. If the integration rule is that a GUI selection overrules other selections, the selection "coffee" is sent to the application. If the integration rule is that speech recognition overrules other selections or that the last selection overrules previous selections, the selection "tea" is sent to the application.

[0023] If no other input is received within the time limit (step 704), the multimodal API forwards, in step 706, the input received in step 701 to the application.

[0024] If the input does not relate to a multimodal event (step 702), the multimodal API forwards, in step 706, the input received in step 701 to the application.

[0025] The difference between these two implementations is described below with a simplified example. Let us assume that an application exists to which multimodal inputs may be given by choosing an alternative from a list shown on a graphical user interface, other inputs are single modality inputs requiring no integration. The alternative may be chosen by selecting it by a mouse click or by giving a spoken selection of a text box or by combining both ways. When a spoken input is received, the corresponding modality API forwards the input to the multimodal API. The multimodal API according to the first implementation described in Figure 6 recognizes whether or not the spoken input is a selection of an alternative on the list and if the input is a selection, the input is forwarded to the "mouse click" modality, otherwise it is forwarded to the application. The multimodal API according to the second implementation described in Figure 7 recognizes whether or not the spoken input is a selection of an alternative on the list and if it is a selection, the multimodal API waits for a predetermined time for an input from the "mouse click" modality, and if the other input is received, combines the inputs and sends one input to the application; otherwise the received spoken input is forwarded to the application.

[0026] In yet another embodiment of the invention, the integrator mechanism described in Figure 7 may be implemented with aspect-oriented programming described in context of Figure 6. This embodiment is referred to as multimodal integrator with aspect-oriented programming. The multimodal API according to this embodiment acts as follows: When a spoken input is received, the corresponding modality API forwards the input to the multimodal API. The multimodal API according to the first implementation described in Figure 6 recognizes whether or not the spoken input is a selection of an alternative on the list, and if the input is a selection, the input is forwarded to the "mouse click" modality; otherwise it is forwarded to the application. The multimodal API waits for "mouse click" modality to response to the input and after receiving response from the "mouse click" modality, the multimodal API forwards the result to the requesting application. It is to be understood that multimodal API may also provide the requesting application with many other types of information.

[0027] Figure 8 illustrates the multimodal API according to a second exemplary embodiment of the invention in which the multimodal API is provided by one class or a package of classes. A multimodal API 8-3 according to

the second exemplary embodiment comprises one or more sets of rules 8-31 (only one is illustrated in Figure 8), registering means 8-32 and listening means 8-33.

[0028] The multimodal API 8-3 may contain a universal set of rules, or the set of rules may be application-specific or multimodal-specific, for example. However, a set of rules 8-31 contains one or more integration rules. A rule may be a predefined rule or a rule defined by an application developer during application designing, or an error-detecting rule defining itself on the basis of feedback received from the application when the application is used, for example. Furthermore, rules and sets of rules may be added whenever necessary. Thus, the invention does not limit the way in which a rule or a set of rules is created, defined or updated; neither does it limit the time at which a rule is defined. The set of rules here also covers implementations in which, instead of sets of rules, stand-alone rules are used.

[0029] The registering means 8-32 and the listening means 8-33 are means for detecting different inputs, and the detailed structure thereof is irrelevant to the present invention. They may be any prior art means or future means suitable for the purpose.

[0030] Figure 9 is a flow chart illustrating a simplified example of how an application developer can create an application utilizing the multimodal API according to the second exemplary embodiment of the invention. The application here is, again, a multimodal user interface, such as a mobile information device applet (MIDlet). First, the application developer selects one or more suitable GUI frameworks (step 901) and one or more modality APIs (step 902). The application developer then selects, in step 903, one or more suitable classes for each selected GUI framework and for each selected modality API. The application developer may have selected a text box implemented by LCDUI and a speech recognizer implemented by JSAPI. The application developer then selects, in step 904, a suitable set(s) of rules or a suitable stand-alone rule(s) for multimodal interaction on the basis of the above selections. The application developer may also fine-tune the rules, if necessary. In another embodiment of the invention, the user may define rules, according to which the rules are dynamically modified during interaction. This embodiment may be utilized in a situation, in which the multimodal API deduces from the input that the user is relatively slow, for example. In such a situation, the multimodal API may lengthen the time it waits for input from a second modality. Finally, the

application developer forms, in step 905, the required interaction on the basis of the above selections (steps 901-904), and the application is ready.

[0031] An example of how the application developer may create an application using the multimodal API according to the second exemplary embodiment of the invention is illustrated by the pseudocode in Figure 10. The pseudocode is based on a J2ME/MIDP LCDUI graphical UI and a JSAPI 2.0 speech API. In the pseudocode of Figure 10, the multimodal API, referred to as an integrator, integrates a speech and a mouse input, i.e. two different ways to select an option from a text box. Section 1001 illustrates a selected modality API(s) and GUI framework(s), section 1002 selecting their classes, section 100 and section 1003 setting an integration rule.

[0032] Although it has been stated above that the application developer selects the set(s) of rules or stand-alone rule(s), the embodiment is not limited to such a solution. The set(s) of rules or stand-alone rule(s) or some of them may be selected by the application.

[0033] Figure 11 is a flow chart illustrating with a simplified example a second implementation of the multimodal API according to the first exemplary embodiment. Also here it is assumed, for the sake of clarity, that the application may receive inputs from two different modalities.

[0034] Figure 11 starts, when the multimodal API listens, in step 1101, events and results from the modalities. In other words, the multimodal API waits for inputs from modalities. An input from a modality API 1 is then received in step 1102. In response to the received input, the multimodal API checks, in step 1103, whether the input relates to a multimodal event. If it relates to a multimodal event, the multimodal API waits, in step 1104, for an input from the other modality API, modality API 2, and a time limit defined by the selected rule set. If the other input is received (step 1105) within the time limit, the multimodal API integrates, in step 1106, the inputs into one input, and sends the input to the application in step 1107. The example of an integration rule disclosed above in Figure 7 may also be applied here.

[0035] If no other input is received within the time limit (step 1105), the multimodal API forwards, in step 1107, the input received in step 1102 to the application.

[0036] If the input does not relate to a multimodal event (step 1103), the multimodal API forwards, in step 1107, the input received in step 1102 to the application.

[0037] The functionality of the second exemplary embodiment is illustrated with a simplified example in which multimodal inputs may be given by choosing an alternative from a list shown on a graphical user interface, other inputs are single modality inputs requiring no integration. The alternative may be chosen by selecting it by a mouse click or by giving a spoken selection of a text box or by combining both ways. When a spoken input is received, the corresponding modality API forwards the input to the multimodal API. The multimodal API according to the second exemplary embodiment recognizes whether or not the spoken input is a selection of an alternative on the list and if it is a selection, the multimodal API waits for a predetermined time for an input from the "mouse click" modality and if the other input is received, combines the inputs and sends one input to the application; otherwise the received spoken input is forwarded to the application.

[0038] Although the embodiments and implementations have been illustrated above with two different modalities, it is apparent to a person skilled in the art how to implement the invention with three or more different modalities.

[0039] The steps shown in Figures 4, 6, 7, 9 and 11 are in no absolute chronological order, and some of the steps may be performed simultaneously or in an order differing from the given one. Other functions can also be executed between the steps or within the steps. Some of the steps or part of the steps can also be omitted. For example, if the modality APIs can themselves recognize whether or not an input relates to multimodal action and, on the basis of the recognition, send the input either directly to the application or to the multimodal API, steps 602, 702 and 1103 can be omitted. Another example relating to applications requiring multimodal inputs is that if no other input is received within the time limit, the multimodal API sends the application an input indicating that an insufficient input was received, instead of forwarding/sending the received input.

[0040] Below, a module and a device containing a multimodal API will be described in general. Detailed technical specifications for the structures described below, their implementation and functionality are irrelevant to the present invention and need not be discussed in more detail here. It is apparent to a person skilled in the art that they may also comprise other functions and structures that need not be described in detail herein. Furthermore, it is apparent that they may comprise more than one multimodal API.

[0041] Figure 12 is a block diagram illustrating a module 120 according to an embodiment of the invention, the module preferably being a software module. The module contains one or more interfaces for inputs 12-1, one or more interfaces for outputs 12-2 and a multimodal API 12-3 according to the invention, such as those described above, for example. The module may be an applet type application downloadable to different devices over an air and/or a fixed connection, or a software application or a computer program product embodied in a computer-readable medium. In other words, the software module may be described in the general context of computer-executable instructions, such as program modules.

[0042] Figure 13 is a block diagram illustrating a device 130 according to an embodiment of the invention. The device contains two or more different modality APIs 13-1 for inputs and interfaces 13-2 for output(s). Furthermore, the device contains one or more applications 13-4 and one or more multimodal APIs 13-3, such as those described above, for example, the multimodal API integrating multimodal inputs for the application or applications. Alternatively, or in addition, the device may comprise the above-described module. The implementation of the device may also vary according to the specific purpose to which the present invention is applied and according to the embodiment used.

[0043] The system, modules, and devices implementing the functionality of the present invention comprise not only prior art means but also means for integrating inputs from two or more different modalities. All modifications and configurations required for implementing the invention may be performed as routines, which may be implemented as added or updated software routines, application circuits (ASIC) and/or programmable circuits, such as EPLD (Electrically Programmable Logic Device) and FPGA (Field Programmable Gate Array). Generally, program modules include routines, programs, objects, components, segments, schemas, data structures, etc. which perform particular tasks or implement particular abstract data types. Program(s)/software routine(s) can be stored in any computer-readable data storage medium.

[0044] It will be obvious to a person skilled in the art that as technology advances the inventive concept can be implemented in various ways. The invention and its embodiments are not limited to the examples described above but may vary within the scope of the claims.

CLAIMS

1. A method for providing interaction between modalities, the method comprising at least:
 - receiving at least one input from at least one modality;
 - manipulating the at least one input according to at least one rule concerning at least one modality; and
 - sending the result of the manipulation to at least one of the group comprising at least one other modality and an application.
2. The method of claim 1, in which the aspect-oriented programming is utilized in manipulating the at least one input.
3. The method of claim 1, in which a multimodal integrator is utilized in manipulating the at least one input.
4. The method of claim 1, in which multimodal integrator with aspect-oriented programming is utilized in manipulating the at least one input.
5. The method of claim 1, in which the at least one rule is manipulated according to input from the at least one modality.
6. A module for providing interaction between modalities, the module being capable of receiving inputs from at least two different modalities, the module comprising at least
 - means for manipulating at least one input received from at least one modality according to at least one rule concerning at least one modality; and
 - means for sending the result of in the manipulation to at least one of the group comprising at least one other modality and an application.
7. The module as claimed in claim 6, wherein the module comprises at least one aspect performing said manipulation.
8. The module as claimed in claim 6, wherein the module comprises at least two aspects chained to perform said manipulation.
9. The module as claimed in claim 6, wherein the module comprises at least one rule defining how said manipulation is performed.
10. The module as claimed in claim 6, wherein the at least one rule is manipulated according to said input from the at least one modality.
11. A computer program product for providing interaction between modalities, said computer program product being embodied in a computer-readable medium and comprising program instructions, wherein execution of said program instructions cause the computer to

obtain at least one input from at least one modality;
manipulate at least one input according to at least one rule concerning at least one modality; and
send the result of the manipulation to at least one of the group comprising at least one other modality and an application.

12. The computer program product as claimed in claim 11, in which the aspect-oriented programming is utilized in manipulating the at least one input.

13. The computer program product as claimed in claim 11, in which the at least one rule is manipulated according to input from the at least one modality.

14. An electronic device capable of providing interaction between modalities, the electronic device being configured at least to
receive at least one input from at least one modality;
manipulate the at least one input according to at least one rule concerning at least one modality; and
send the result of manipulation to at least one of the group comprising at least one other modality and an application.

15. The electronic device as claimed in claim 14, in which the aspect-oriented programming is utilized in manipulating the at least one input.

16. The electronic device as claimed in claim 14, wherein the electronic device comprises at least one aspect performing said manipulation.

17. The electronic device as claimed in claim 14, wherein the integrator is configured to recognize whether or not an input relates to a multimodal interaction, and in response to the input not relating to a multimodal interaction, to forward the input directly to the application.

18. The electronic device as claimed in claim 14, in which the at least one modality is selected from a group of a mouse, a keyboard, a stylus, speech recognition, gesture recognition, haptics recognition, input from an in-car computer, distance meter, navigation system, cruise control, thermometer, hygrometer, rain detector, weighing appliance, timer and machine vision.

19. An application development system comprising at least one framework, at least one modality application programming interface and at least one multimodal application programming interface, the system providing means for at least

receiving at least one input from at least one modality;

manipulating the at least one input according to at least one rule concerning at least one modality;

sending the result of the manipulation to at least one of the group comprising at least one other modality and an application.

20. The application development system as claimed in claim 19, wherein said multimodal application programming interface is provided by at least one aspect comprising at least one rule.

21. The application development system as claimed in claim 19, wherein said multimodal application programming interface is provided by a set of rules, the system further comprising selection means for selecting, for an application, at least one framework, at least one modality application programming interface and at least one rule from the set of rules.

22. The application development system as claimed in claim 19, in which the aspect-oriented programming is utilized in manipulating the at least one input.

1/7

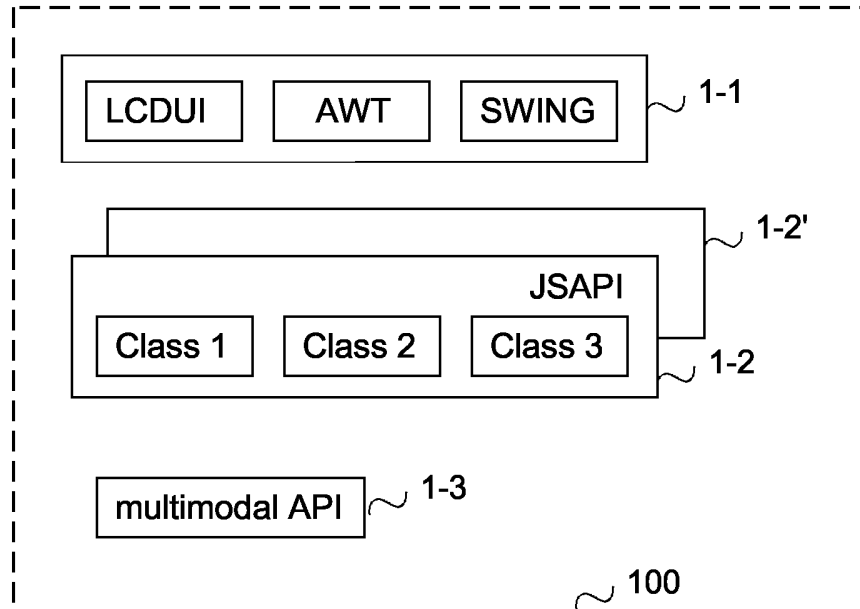
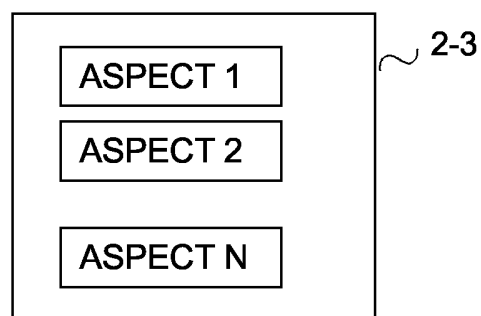
*FIG. 1**FIG. 2*

FIG.3A

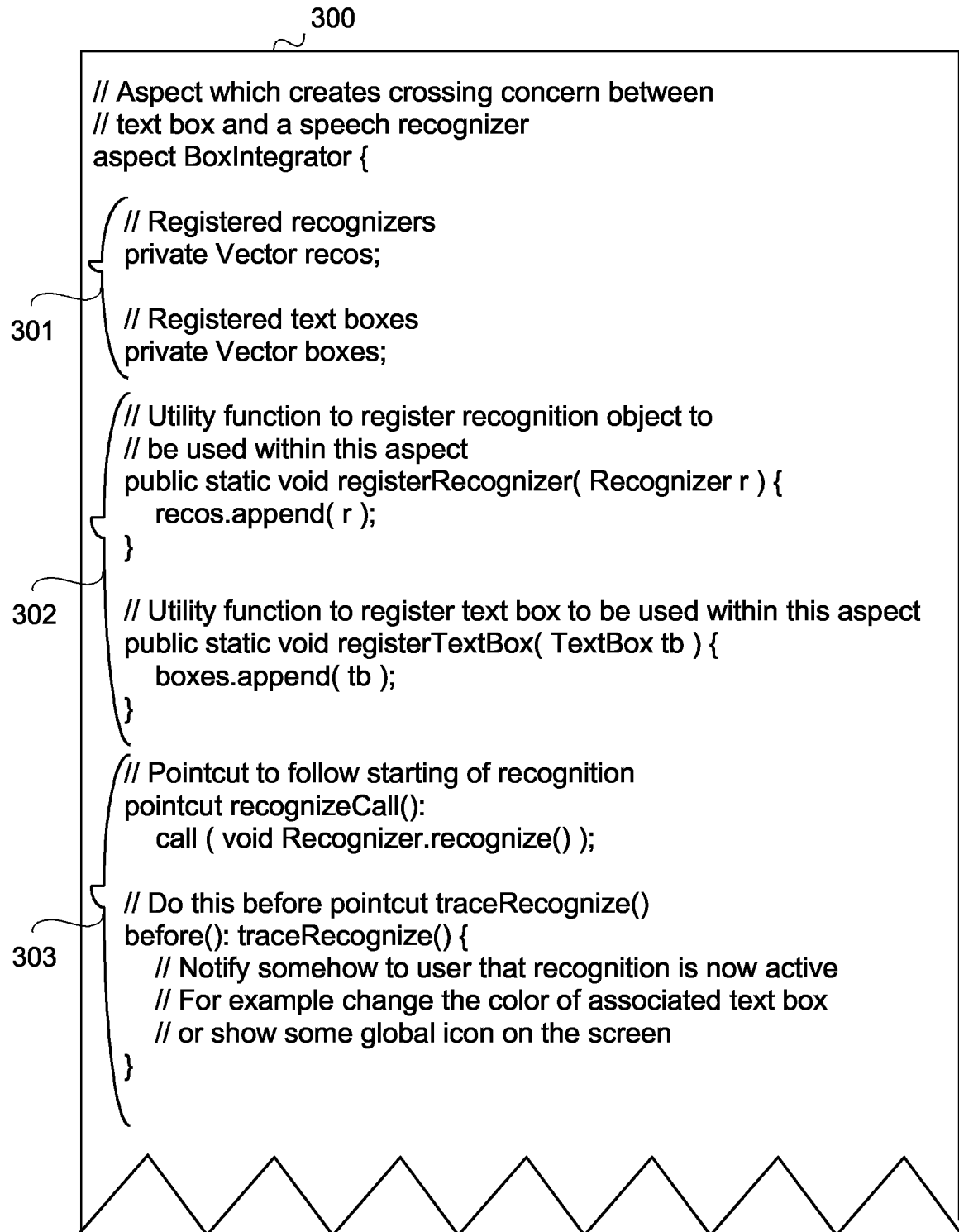
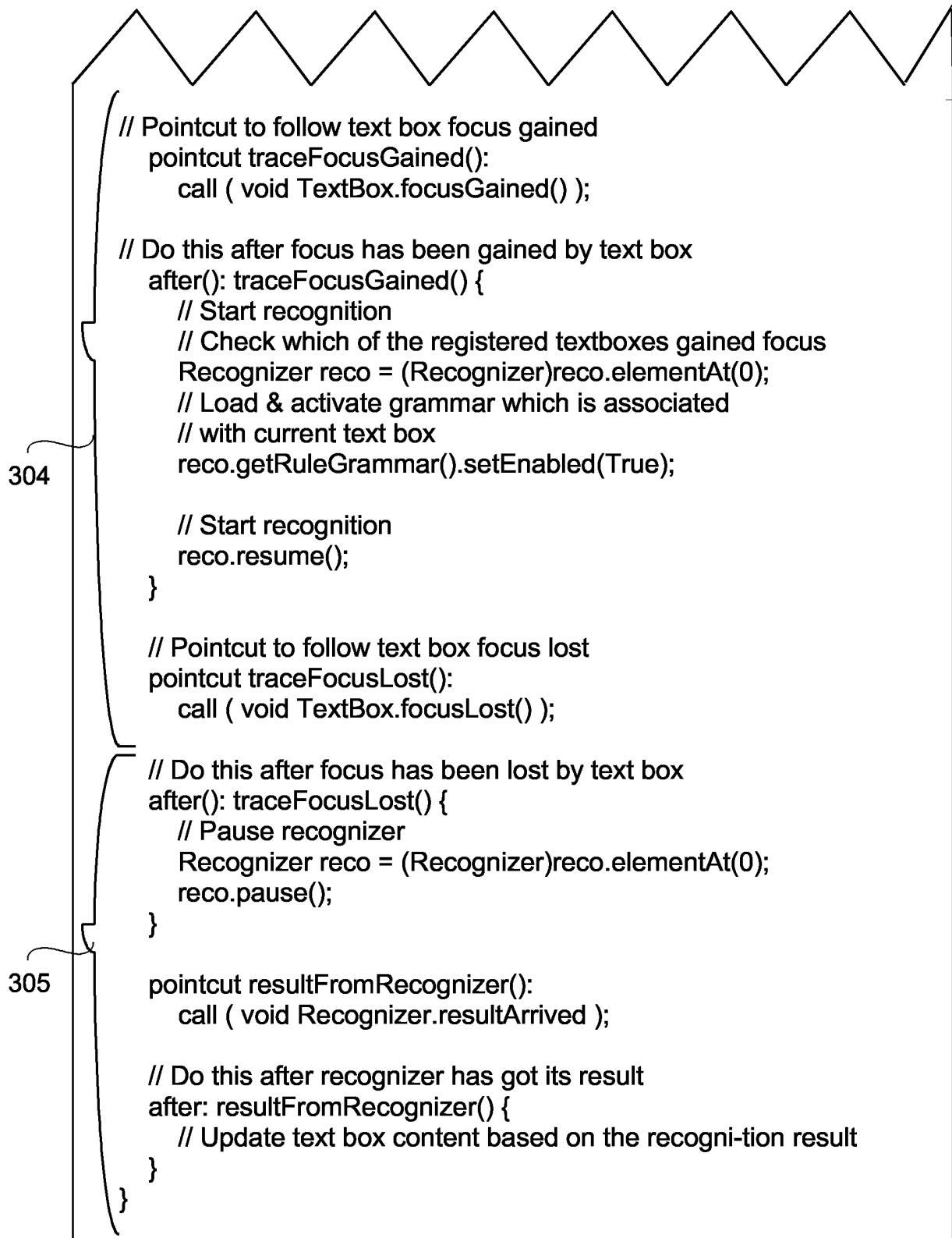
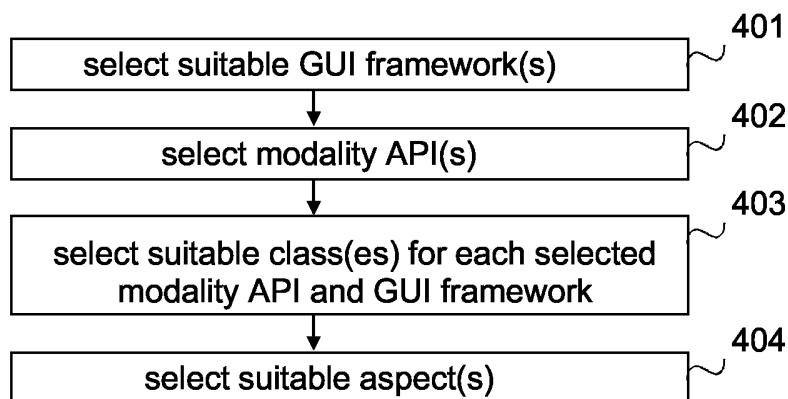
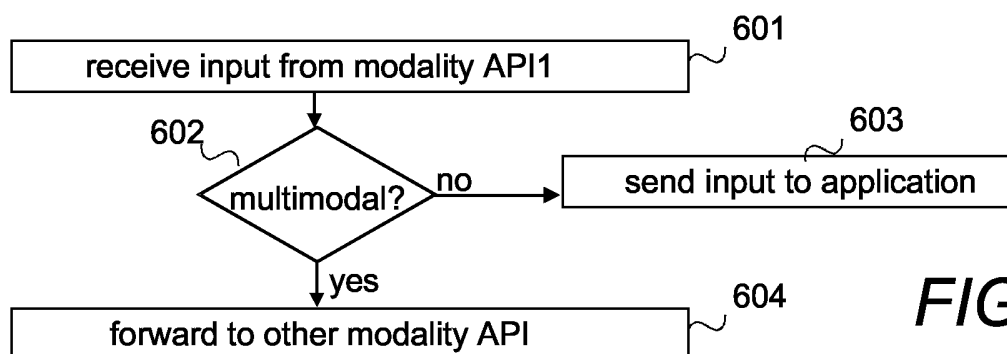
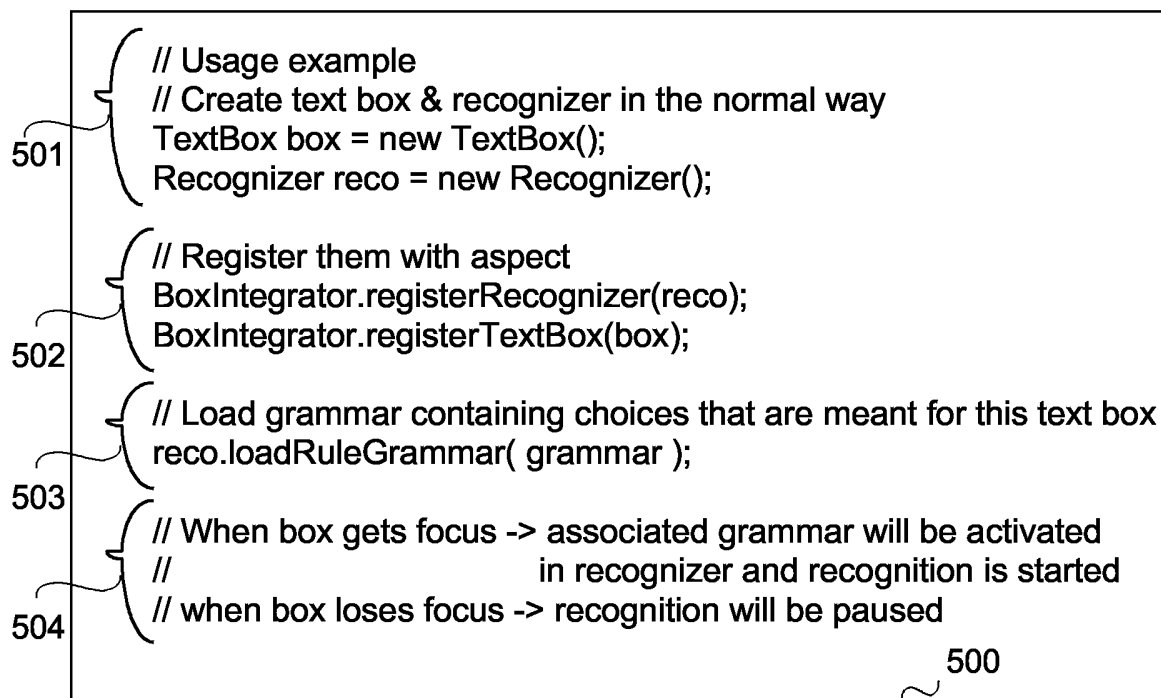
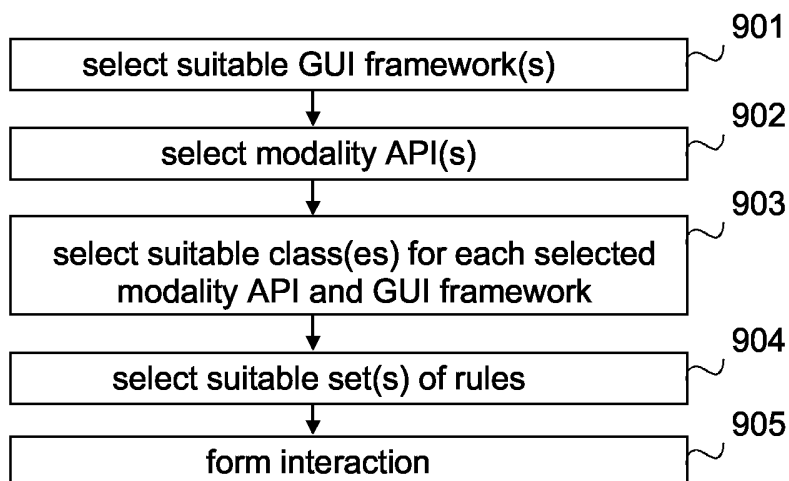
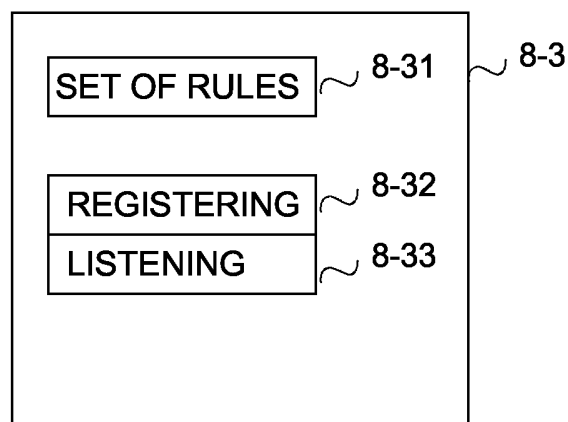
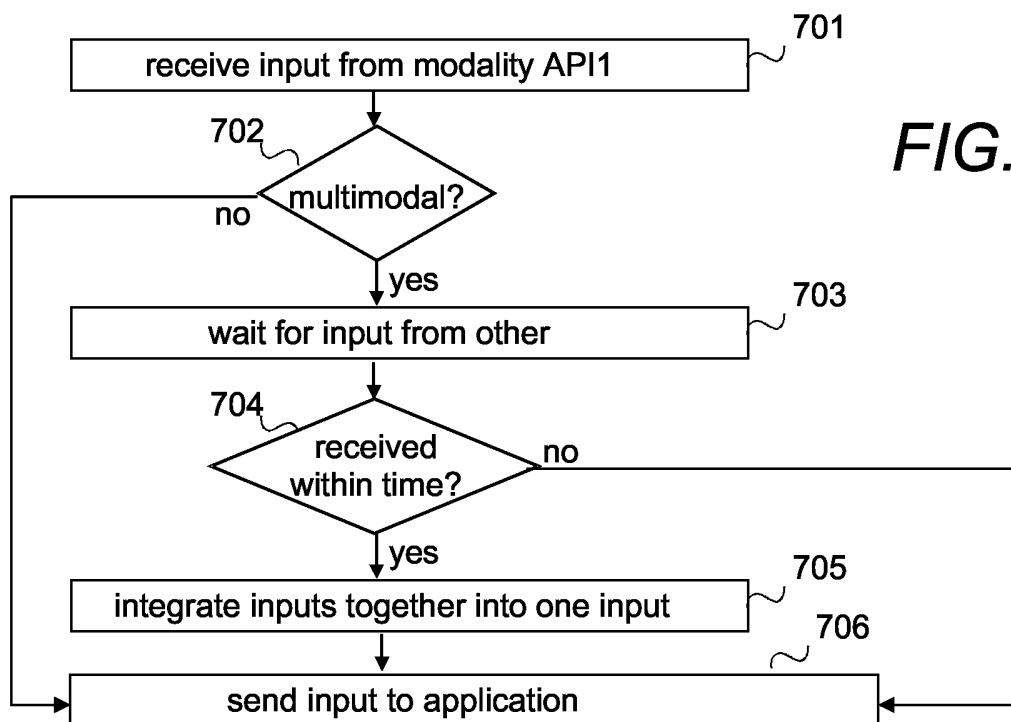


FIG.3B

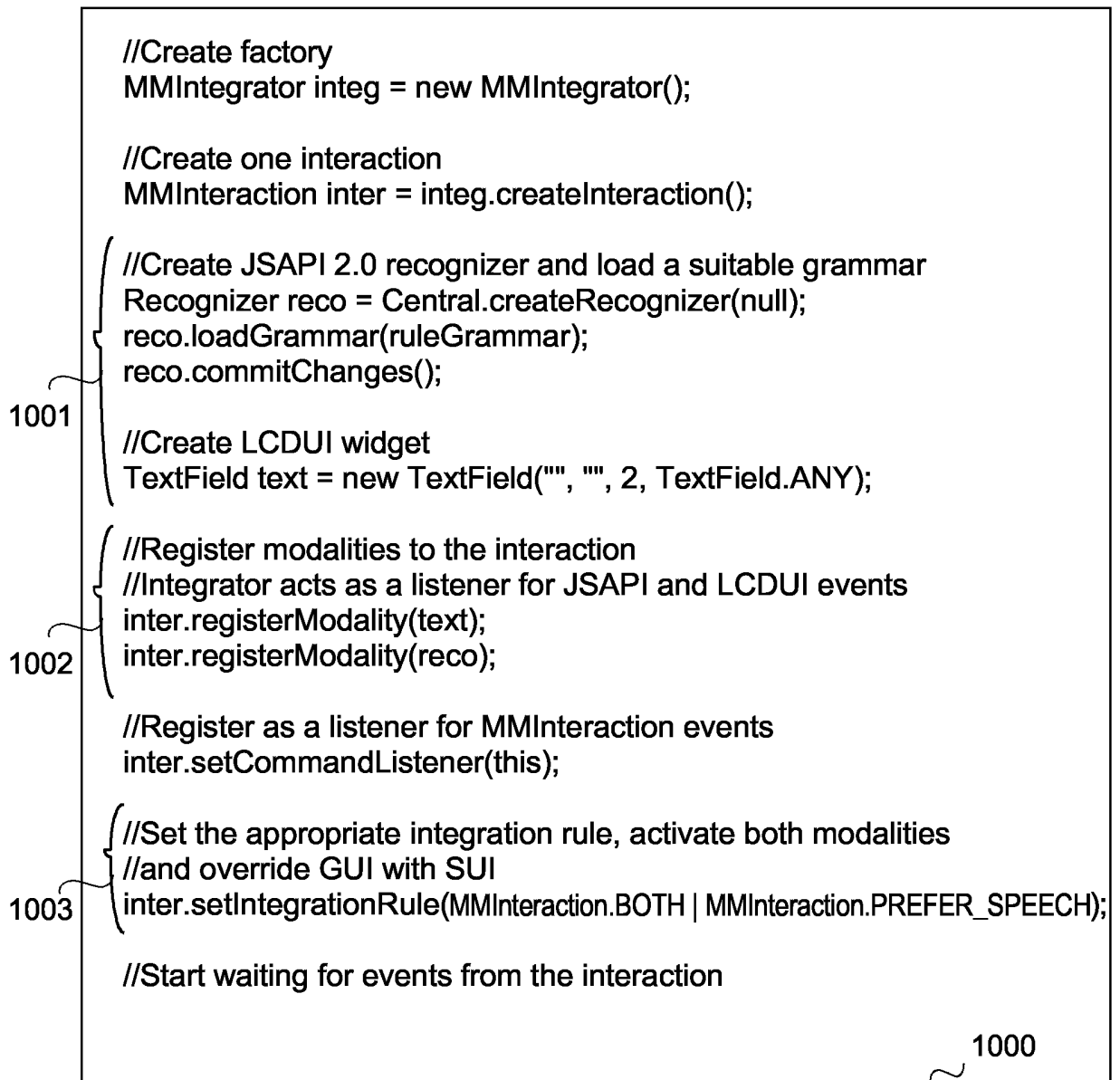
4/7

**FIG. 4****FIG. 5****FIG. 6**

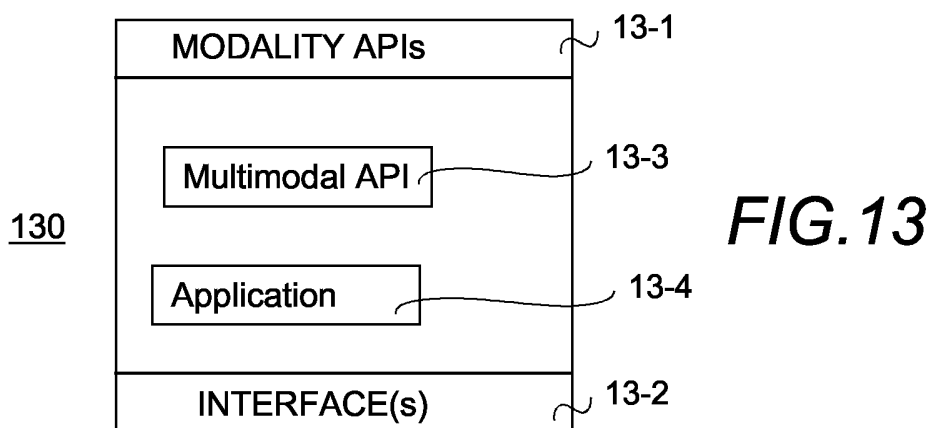
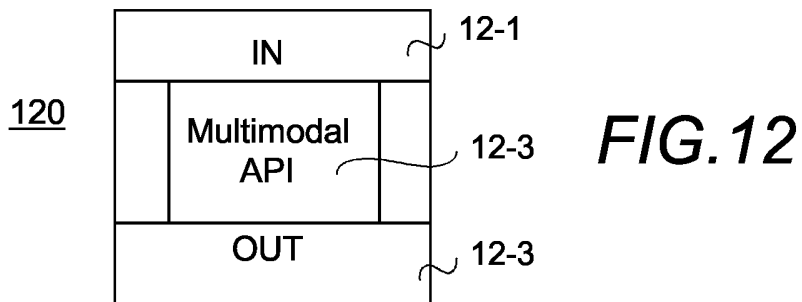
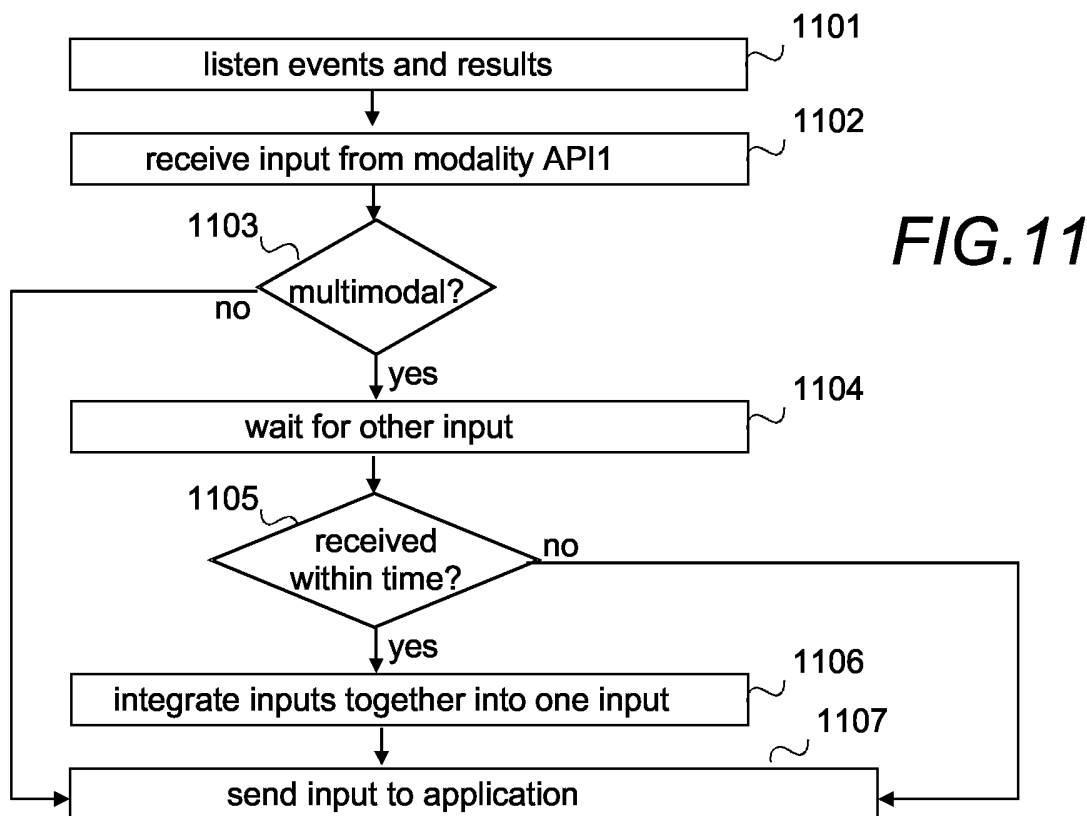
5/7



6/7

**FIG. 10**

7/7



INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2005/050487

A. CLASSIFICATION OF SUBJECT MATTER		
See extra sheet		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
IPC 8: G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
FI, SE, NO, DK		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
EPO-Internal, WPI, PAJ, Inspec, XPI3E, XPESP		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Flippo, et al, 'A framework for rapid development of multimodal interfaces', ICMI'03, Nov 5-7, printed from: http://web.archive.org/web/*/http://www.caip.rutgers.edu/disciple/Publications/icmi2003.pdf , public 30 March 2004,, whole document; chapter 3.1 in particular	1-22
X	Cohen, et al, 'QuickSet: multimodal interaction for distributed applications', ACM Multimedia#97, Seattle, 1997, whole document	1-22
X	Trabelsi, et al, 'A voice and ink XML multimodal architecture for mobile e-commerce systems', WMC#02, 28 September, Atlanta, 2002, whole document	1-22
X	Barthelmess, et al, 'Aspect-oriented Composition in Extensible Collaborative Applications', In Proc. 2002 Int. Conf. on Parallel and Distributed Processing Techniques and Appl. (PDPTA02), Las Vegas, June 2002, whole document; chapter 3, in particular	1-22
☒ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
16 March 2006 (16.03.2006)		24 March 2006 (24.03.2006)
Name and mailing address of the ISA/FI National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland Facsimile No. +358 9 6939 5328		Authorized officer Olli-Pekka Piirilä Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2005/050487

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 2001075838 A (KOREA ELECTRONICS TELECOMM) 11 August 2001 (11.08.2001)	
A	US 2002/0194388 A1 (BOLOKER DAVID et al.) 19 December 2002 (19.12.2002)	
A	Kiczales et al, 'Aspect Oriented Programming', in Proc. European Conf. On Object-Oriented Programming (ECOOP#97), 1997	
A	US 6467086 B1 (KICZALES GREGOR J et al.) 15 October 2002 (15.10.2002)	
A	US 6539390 B1 (KICZALES GREGOR J et al.) 25 March 2003 (25.03.2003)	
A	US 2003/0149959 A1 (LAMPING JOHN O) 07 August 2003 (07.08.2003)	
A	W3C Multimodal interaction framework, http://www.w3.org/TR/2003/NOTE-mmi-framework-20030506/ , public 23 May 2003, see http://web.archive.org/web/*/http://www.w3.org/TR/2003/NOTE-mmi-framework-20030506/	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2005/050487

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
KR 2001075838 A	11/08/2001	None	
.....			
US 2002/0194388 A1	19/12/2002	US 2005273759 A1 US 2002198719 A1 US 2003023953 A1	08/12/2005 26/12/2002 30/01/2003
.....			
US 6467086 B1	15/10/2002	CA 2313721 A1 EP 1074911 A2	20/01/2001 07/02/2001
.....			
US 6539390 B1	25/03/2003	None	
.....			
US 2003/0149959 A1	07/08/2003	JP 2003233499 A	22/08/2003
.....			

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2005/050487

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.

G06F 3/01 (2006.01)