Robot-Type Integrated User Interface Platform

TAKANO Yosuke, YAMASHITA Nobuyuki, FUJITA Yoshihiro

Abstract

Advances in the functions of terminals including cellular phones and car navigation systems have made an easy-to-use UI (User Interface) and a platform supporting it more required than ever. To meet these needs, NEC has developed a UI platform for the implementation of a human-friendly robot, called RoboStudio, which we are currently trying to apply laterally to various terminals. This paper is intended to describe the actual measures that are being taken to achieve this result, including Linux implementation, weight reduction and CG rendering.

Keywords

user interface (UI), robot, platform, embedded equipment

1. Introduction

At NEC, we are conducting R&D into personal partner-type robots that will live together with humans in future households. We have particularly focused on a compact communication robot that features speech and vision recognition functions, can play with humans, can talk and can support the operation of home electrical appliances and access Internet information. After succeeding in prototyping a personal robot called PaPeRo in 2000, we have been carrying out improvements in preparation for its practical application.

PaPeRo offers an interface that goes one step beyond that of traditional terminals that wait until a button is pressed, for example proposing the use of a function according to the user’s skill or detecting a command error and giving guidance on correct command usage. Our idea was to develop the UI technology via robots by applying it to various consumer and professional equipment in order to eventually implement an “understanding,” “perceptive” interface that can be used “safely,” “confidently” and “comfortably” by anyone in the Ubiquitous Society. In addition, we also believe from the viewpoint of developers that the robot technology is effective for UI implementation. In the future, it is expected that the terminal equipment will incorporate the means for collecting environmental and user information using various sensors such as an acceleration sensor and GPS, as well as various means of representation for advanced and multiple functions with user-friendly methods for delivery to users. Since these various input/output configurations are identical to those in current use with robots, we believe that the technology for the configuration of software incorporated in the robot (platform) will also be applicable to the implementation of the UI for a variety of terminals. The technology for robot configuration specifically means the platform for processing a large number of inputs (events) and the environment for facilitating the development of the input/output control logic.

We named the targeted UI as the robot-type UI and, based on the knowledge we acquired, enumerated three points as the requirements for the robot-type UI.

(1) Multimodality (N inputs, N outputs)

The UI should interpret the will and problems of users based on multiple input means and inform the user effectively of information using multiple output means including voice and gestures.

(2) Autonomy

The UI should not simply wait for commands from the user, but should be capable of anticipating the intentions of the user by proposing functions, etc.

(3) Intelligence

The UI should learn changes in user behavior and environment and reflect these in performance.

As the first step toward the implementation of the robot-type UI, we developed a robot software platform called RoboStudio as the PaPeRo execution/development environment. We are currently trying to apply this laterally to various terminals other than the robot. In the following section, we will describe these attempts.

2. Robot Software Platform “RoboStudio”

RoboStudio is the development environment/execution system for robot software that supports the robot’s human communication capability. Its components are based on the mechanisms and knowledge that we have gained from R&D into the personal robot PaPeRo. RoboStudio features applicability to any kind of robot and the provision of the entire software systems required for the configuration of a robot, from recognition technologies to the language for describing the robot’s behavior. As shown in **Fig. 1**, RoboStudio is composed of; 1) workers (components) including the speech and face image recognition functions that can be used in a real environment; 2) a robot virtual machine that is composed of the messaging mechanisms between the scenario interpreter and the components; 3) a development tool for robot-dedicated application called the scenario/motion tool; 4) a database processing function that supports communications.

3. Measures for Application to Terminal Equipment

In order to enable application of RoboStudio to a variety of terminal equipment, we suggest two solutions; 1) Linux implementation and weight reduction, and; 2) CG rendering.

3.1 Linux Implementation and Weight Reduction

RoboStudio was originally implemented using Windows as the base OS, because the rich line of media processing libraries of Windows was suitable for use with a robot that had the objective of interaction. However, we found that transition to an OS with a widely scalable range and a compatibility with various processors is necessary in order to embed in small-sized equipment that has a small number of resources. Therefore, we decided to port it to Linux and also because this was advantageous from the viewpoint of the license fee. Up to the present, we have ported RoboStudio to several types of Linux on several types of processor (Pentium, ARM9).

In porting to the ARM9 processor, the cost of execution of the robot VM of RoboStudio posed a problem. Message communications between scenarios and workers are generated very frequently in RoboStudio. For example, 50 to 100 messages are communicated during a gesture of PaPeRo.

We finely analyzed and reduced the cost required for each operation in message communications, including the generation, queuing and configuration information management, and eventually succeeded in reducing the delay time related to message exchanges to less than 1/10th of the Windows version.

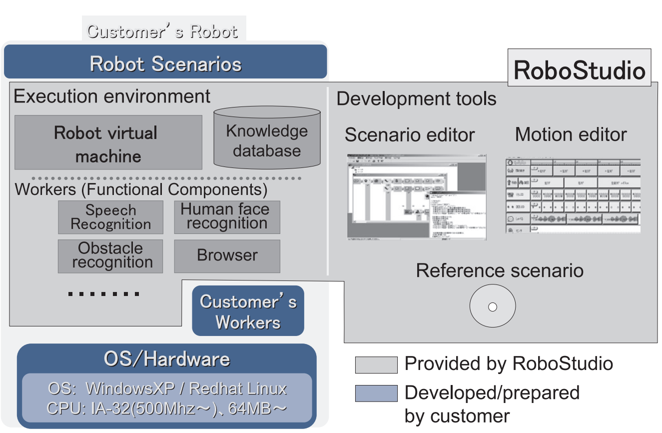

Fig. 1 Configuration of RoboStudio.

3.2 CG Rendering

Although PaPeRo has the means of output using voices and mechanical operations, we thought it effective to use graphics on the terminal display as one means of output in order to implement the robot-type UI on a variety of terminals. For this purpose, we incorporated the following two means in RoboStudio of rendering graphics (3D, 2D), which can either be used together or selected according to a purpose. We believe that this will allow the robot-type UI to provide the required multimodal expressions.

(1) 3D CG Rendering Function (CG Worker)

A controllable robot is drawn in 3D CG on the display (**Photo 1**). Projection of the character agent on the display screen makes it possible to transmit information to the user by the same actions and gestures as the real robot. For this purpose, we obtained cooperation from HI Corporation and incorporated its MascotCapsule engine as a worker (function module).

(2) Flash Rendering Function (Flash Worker)

The 2D representation created by Flash contents can be controlled from a scenario. It can also be used to draw the background of the 3D-drawn robot as shown in Photo 1. We implemented this by calling up the Adobe Flash player via ActiveX.

In addition, we provided the CG worker with the capability of controlling the actions of CG characters using the same interface as that used in processing the actions of the real robot, thereby making it possible to develop applications that can

Photo 1 CG rendering of PaPeRo.

work with both the real robot and CG character. When this is applied, it is also possible to implement a service that lets the CG character on a portable terminal take over the interactive processing performed with the real robot.

4. Packaging the Speech Recognition Module

We developed a hardware module that contains RoboStudio featuring Linux implementation and CG rendering as described above as the core and can provide the services required for various ubiquitous equipment, and named it the “Speech Recognition Module.” Part of this development was conducted as an entrusted operation of the “Project for the Development of a Common Platform for a Next-Generation Robot” of NEDO (**Photo 2**).

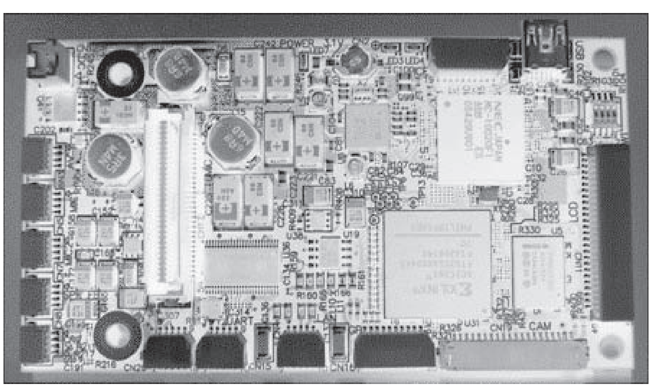

Photo 2 Speech recognition module.

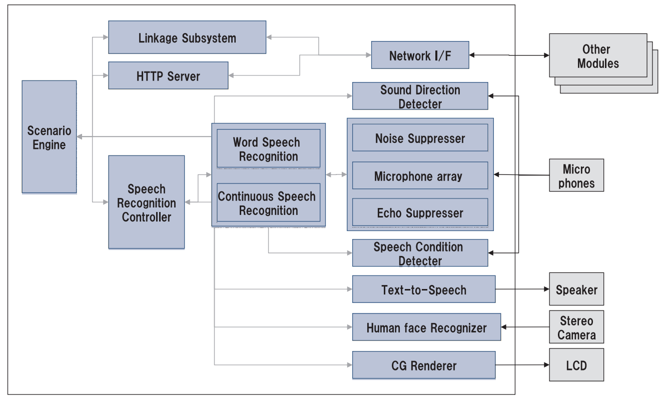

Fig. 2 Configuration of voice recognition module.

The hardware of the Speech Recognition Module has a similar size to a business card and includes the MP211 multi-core processor (which is composed of three ARM9 192MHz processors, a DSP processor and a 128MB memory that is packaged as the SIP) of NEC Electronics Corporation as the core, as well as various peripheral interfaces that are compatible with the 16CH microphone inputs, camera input, voice output and LCD output. The hardware also mounts a speech noise suppresser, speech recognizer (word recognition engine and large-vocabulary continuous recognition engine), human face recognizer, text-to-speech function, CG rendering engine, network function, scenario engine, etc. These functions in the Speech Recognition Module can be started up selectively.

Fig. 2 shows the block diagram of the software functions of the Speech Recognition Module. Incorporating the Speech Recognition Module as a part of the ubiquitous equipment, it will be possible to build ubiquitous equipment with the robot-type UI. Among the potential functions, we have already confirmed the operation of two worker functions, which are word speech recognition and text-to- speech.

5. Conclusion

In the above, we discussed the robot-type UI project and the measures taken for the application of the RoboStudio software platform to various terminals as the first step of development. In the future, we will advance the fulfillment of the three requirements for the robot-type UI at the same time as pioneering specific applications that are related to cellular phone and car-mount terminals.

* Windows, Internet Explorer and ActiveX are trademarks or registered trademarks of Microsoft Corporation in the USA and other countries.
* Linux is a trademark or registered trademark of Linus Torvalds in the USA and other countries.
* ARM is a registered trademark of ARM, Ltd, in the EU and the USA.
* MascotCapsule is a registered trademark of HI Corporation in Japan.
* Flash is a trademark or registered trademark of Adobe Systems, Incorporated in the USA and other countries.

Authors' Profiles

TAKANO Yosuke
Principal Researcher,
Common Platform Software Research Laboratories,
NEC Corporation

YAMASHITA Nobuyuki
Principal Researcher,
Common Platform Software Research Laboratories,
NEC Corporation

FUJITA Yoshihiro
Senior Manager,
Enterprise Solutions Planning Division,
NEC Corporation