Application Functions for Winter/Spring 2010-2011 Models
—Evolving Mobile Terminal Applications—

As functions and services provided by mobile terminals become increasingly diverse and complex, there is a growing need for improving user-behavior support and terminal operability while also providing functions and services that satisfy users who are always seeking new experiences. The NTT DOCOMO winter/spring 2010-2011 models feature built-in applications that provide more extensive support of user behavior and terminal operations while continuously providing users with new experiences.

1. Introduction

NTT DOCOMO has been providing user-behavior and user-lifestyle support on its mobile terminals through services like i-concier and iBodymo™. However, the functions and services provided by mobile terminals are becoming increasingly diverse, and the operations that the user needs to perform to reach and start them up are becoming all the more complicated. Thus, to get users to make more effective use of these functions and services, it is imperative that the operability of the mobile terminal itself as well as the process flow to start up functions and services be improved. Additionally, the spread of high-performance and high-function portable game consoles, smartphones, and other digital devices requires that mobile terminals also respond to the needs of users who seek more sophisticated experiences.

We have developed new applications for NTT DOCOMO’s winter/spring 2010-2011 models (Photo 1) to meet this wide array of needs on the user level. These applications aim to provide rich user experiences on an ongoing basis while providing thorough support for user behavior and terminal operations.

In this article, we describe the i-concier function with extended input-support and remind functions, the Machi-chara function with extended speech dialogue and network-update functions, and an i-appli that extends the music player function.

2. i-concier Function Extensions

2.1 Background to Development

Since its launch in the autumn/winter 2008 models, the i-concier service has been extended with various functions such as expanded capacity for data stored on the i-concier server and information delivery linked with user location information. Now, in the winter/spring 2010-2011 models, NTT DOCOMO has developed extensions to the scheduler application with the aim of storing user intentions in the mobile terminal to provide more detailed assistance and effective user-behavior support.

*1 iBodymo™: A trademark of NTT DOCOMO, INC.
2.2 Data Format Extension

The schedule text-memo and ToDo-list data formats have been integrated by extending their properties based on the vEvent format for schedule data. This extension enables these data (hereinafter referred to as “memo”) to be processed by the scheduler application. It also makes it easy for the user to register and view various types of information without having to worry about application type or input format. The display of memo-creation screens is performed using VIVID UI\(^2\) content thereby providing uniform operability across different terminal models.

2.3 Improved Operability in Data Creation

The following functions were developed to enable user intentions to be easily stored within the mobile terminal.

1) Simple Memo Creation Function

This function enables the user to create a memo starting from the standby screen through simple operations. The Machi-chara function leads the user to the i-concier top screen from which a Front End Processor (FEP)\(^3\), camera, or other function can be directly started up. A memo can therefore be registered by simply inputting text or taking a picture, for example. This function achieves a memo-creation UI based on the concepts of easy operation and “let the mobile terminal store information for you.”

2) Input Support Function for Memo Creation

Information on “when,” “where” and “with whom” that can be set for a memo is usually entered by inputting numbers, symbols, or Japanese language text. The input support function supplements this input by using user data to predict information on “when,” “where” and “with whom” that the user is about to enter and then display a list of candidates. The user then only has to select the correct candidate to input that information. With this function, the user can learn how to make memos with ease.

This function uses morphological analysis to extract nouns from text input in the “subject” and “details” fields of memos, and then displays a list of hints corresponding to any temporal, location or personal information among those nouns in a specific priority order.

The process flow of the input support function is shown in Figure 1. The input-support data extraction component receives text data from the memo application and controls the entire process flow up to the output of a hint list. To begin with, the input-support data extraction component passes text data to the morphological analysis component, which performs morphological analysis on that text. This component then compares those words classified as nouns with information in a standard dictionary and information added to the

---

*2 **VIVID UI**: A user interface platform that can handle various kinds of multimedia content in an integrated manner. “VIVIDUI” is a registered trademark of Acrodea, Inc.

*3 **FEP**: In this article, a kana-based Japanese-language input function.
dictionary, and extracts and returns only those words corresponding to “when,” “where” or “with whom.” Next, the input-support data extraction component takes these words classified as “when,” “where” or “with whom” and passes them to the “when” data extraction component, “where” data extraction component and “with whom” data extraction component, respectively. Each of those components compares the words it receives with other data, switches values around as needed, and rearranges the words in a final output format, and after assigning a priority order, returns them to the input-support data extraction component.

Furthermore, as a new function, users can now create memo from incoming and outgoing e-mail. Here, a wide variety of hints can be extracted from the subject and main body of an e-mail so as to automatically reflect the subject and main body of an e-mail in the subject and details of the memo.

### 2.4 Viewer Customization

The autumn/winter 2008 models introduced graphics in the calendar display for schedule data using VIVID UI content. The winter/spring 2010-2011 models provide more extensions such as a week-by-week display and support a decoration function that enables the pasting of Deco-mail pictograms and decorations on the calendar. These extensions enable the user to customize the calendar display on the mobile terminal in much the same way that one’s personal day planner can be customized to suit one’s needs. Examples of calendar screen shots are shown in Figure 2.

### 2.5 Remind Function

We have developed a remind function using information stored by the user on the mobile terminal to improve user-behavior support.

1) Current-day Schedule Notification Function

This function notifies the user once per day at the time of day specified by the user of the number of events scheduled for the current day. Here, the i-concier event-notification function has been extended to enable the message notifying the user of the current day’s number of events to be generated inside the mobile terminal. It has also been extended to enable a pop-up message linked with Machi-chara content to be displayed on the standby screen and for an event list to be displayed on a dedicated viewer. This pop-up message on the standby screen is also supported by

---

**Figure 1 Process flow of input support function**

**Figure 2 Calendar screen shots**
3. Machi-chara Function Extensions

3.1 Background to Development

As one method for improving the operability of the mobile terminal itself and the operation flow to find and start up a function, the winter/spring 2010-2011 models provide operation support within the mobile terminal using the Machi-chara function. Specifically, we have developed a function for guiding the user to the function that the user would like to start up using a speech dialogue between the user and Machi-chara character. Speech dialogue is achieved by combining a speech recognition function, a text-to-speech function, and a speech dialogue function for controlling the conversation between the user and Machi-chara.

3.2 Speech Dialogue Function

Sequence flow of the speech dialogue function is shown in Figure 3. In the winter/spring 2010-2011 models, the dialogue begins with Machi-chara saying “What function would you like to use?” The user then responds by uttering the name of the desired function. The speech uttered by the user is now converted into text by the speech recognition function and notified to the speech dialogue function. The speech recognition function uses Local Speech Recognition (LSR) [*4] [1] inside the mobile terminal developed for the autumn/winter 2008 models.

---

[*4] LSR: A system that completes all processing for extracting speech features from input speech and converting those features to recognition results inside the mobile terminal using a built-in speech-recognition engine.
The speech dialogue function now compares the text it has received with a list of executable functions described in speech dialogue scenarios that it possesses and decides whether one of those functions has been uniquely specified. At this time, if the user utterance does not completely match the formal function name, the speech dialogue function may guide the user to the function that best fits certain keywords in the utterance or to an explanation of this operation in the existing help function. Support is also given here for sentences uttered in a colloquial manner, sentences with or without particles, and sentences ending with a variety of common expressions so as to minimize restrictions on the user’s speaking patterns as much as possible. If the function desired cannot be narrowed down with one utterance from the user, the user will be asked for clarification in a conversational manner so that additional information can be obtained and the function desired can be identified and started up. Once the desired function is uniquely identified, the function ID is notificated to an upper level application to call up that function.

Questions asked the user or information notificated on to the user follow information described in speech dialogue scenarios. Statements that match the situation in question are notificated from the speech dialogue function to an upper level application so that they can be displayed on a pop-up while being spoken by Machi-chara.

### 3.3 Text-to-speech Function

We have developed a new text-to-speech function to make conversation with respect to user’s input speech more natural. In this function, state-
ments to be read out as described in speech dialogue scenarios are notificat-
ed from the speech dialogue function to the text-to-speech function to generate synthesized speech.

Text-to-speech here adopts a method based on the Hidden Markov Model (HMM)\(^5\). Thus, even without a database of previously recorded and generated speech, any speech can be output by converting HMM parameters. Compared to other text-to-speech methods, HMM can be achieved with a smaller model size making it applicable to use in mobile terminals for which memory is limited.

This method generates synthesized speech based on synthesized-sound models having good speech quality, intonation, and other features. If a synthesized-sound model exists for Machi-chara data, the system will use the specific model accompanying Machi-chara to generate speech, and if there is no synthesized-sound file for the Machi-chara data, or Machi-chara settings are turned OFF, the system will use the synthesized-sound model already possessed by the text-to-speech engine inside the text-to-speech function to generate speech.

3.4 Machi-chara Network Update Function

In the existing Machi-chara function, Machi-chara content must be downloaded by user operations via a browser and saved on the mobile terminal. To make updating easier for the user, we have developed a Machi-chara network update function as a mechanism that can update Machi-chara content even without user operations (Figure 4).

This function actually consists of two functions: an automatic update function requiring no user operations or notifications, and a manual update function in the event that automatic updating fails or the user has no desire for automatic updating.

1) Automatic Update Function

The process flow of the automatic update function is shown in Figure 5. The function starts up when the content provider sends an incoming notification to the mobile terminal at the time of its choosing. The update target is determined at the mobile terminal. To prevent Machi-chara content from being erroneously updated, the incoming notification includes an ID that differs for each set of Machi-chara content (hereinafter referred to as “CFDID”).

To download existing Machi-chara content, the process first gets the MMD file containing Machi-chara metadata and then uses the information contained in that file to get the CFD file having actual Machi-chara content data. The MMD file contains the URL for getting the CFD file as well as other pertinent information such as CFD file size.

The automatic update function uses the CFD file size specified in the MMD file to secure memory in the mobile terminal before attempting to download the CFD file. This prevents a “cannot save” state from occurring during the downloading process. In the event of insufficient memory in the mobile terminal, the function displays a desk-top icon for user-notification purposes before initiating a CFD file download and makes a transition to manual updating.

After completing the CFD-file download after updating, the Machi-chara content settings displayed on the mobile terminal’s screen are automatically changed in Machi-chara content.

2) Manual Update Function

The process flow of the manual update function is shown in Figure 6.

---

*5 HMM: A probability model widely used in the study of speech signals as a technique for statistically modeling a speech spectrum.
This function enables Machi-chara content stored on the mobile terminal to be updated by user operations without having to use a browser. In short, the user need not go through the trouble of starting up the browser to check whether Machi-chara content has been updated.

4. i-appli Music Player Function
4.1 Background to Development

We have introduced an i-appli music player function that enables music data files inside the data-box music folder to be accessed from an i-appli.

In the past, music files could be played back on an i-appli by preparing a musical rendering such as a MIDI file as a resource within the i-appli. However, given an i-appli environment with a memory limit of 2 MB, it was difficult to handle large-capacity music content. This new i-appli music player function makes it possible to provide a music playback function unlimited by i-appli capacity by sharing the music content database managed by the native music player with the i-appli execution environment (Figure 7). Additionally, by combining this function with the individual charging function and full-music-track download function introduced in spring/summer 2009 models, music content can now be purchased, downloaded, saved and played all from an i-appli. This function is expected to provide content providers with a new source of revenue as well as to be combined with other i-appli functions to form mashups*6. For example, adding a music player function to an integrated wellness application like iBodymo that includes functions like a pedometer, calorie counter and route detection navigator should make it possible to provide users with new added value and enjoyable experiences (Figure 8).

The i-appli music player function consists of a music playback control function, music content management function, and Bluetooth® Advanced Audio Distribution Profile (A2DP) and Audio/Video Remote Control Profile.

---

*6 Mashup: In this article, a service formed by combining a variety of APIs.

*7 Bluetooth®: A registered trademark of Bluetooth SIG, Inc. in the United States.
(AVRCP) functions.

4.2 Music Playback Control Function

The music playback control function manages the state of music playback software, issues control requests, gets and sets attribute information, and gets events. Music playback software has six states, and the i-appli issues control requests associated with these state transitions to this software. The relationship between control requests from the i-appli and state transitions of the music playback software is shown in Figure 9. The i-appli can also specify and control play lists, music content and the playback order of songs and it can get and set attribute information required for music playback. This attribute information deals with volume and song playback mode (seven modes in all differing in shuffle/repeat settings). The i-appli music player function is also expected to save on energy during music playback, and to this end, we have made it possible to suspend only the i-appli while music is playing. For example, the i-appli can be suspended while the music playback software is playing a song except when user operations or screen updates are

![Image](http://example.com/image.png)

*This is provided only in Japanese at present.*

**Figure 8** Example of an i-appli using the i-appli music player function

![Image](http://example.com/image.png)

**Figure 9** Control requests and state transitions in music playback software
needed. The i-appli must be able to
detect changes in the state of the music
playback software at all times, even
when it is starting up and returning from
a suspended state. We have therefore
equipped the music playback software
with a function to notify the i-appli of
events whenever its state or settings
change. In addition to events associated
with the various types of state transitions,
there is also an event corresponding to
song switching during playback and an
event indicating the need for updating
the music content database as well as a
variety of error events.

4.3 Music Content Management
Function

The music content management
function enables artist, album or genre
to be specified from the i-appli as a filter
to get music content from the music
content database managed by the native
music player. It also enables the i-appli
to get information associated with
music content like song name, artist
name and jacket graphics. In this
regard, the native music player man-
ages music content stored on the mobile
terminal itself as well as data inserted in
the mobile terminal on the content of
external storage media. This means that
the i-appli can manage music content
without having to be aware of where it
is being stored.

4.4 Bluetooth A2DP and AVRCP
Functions

To respond to user needs like “I
want to hear music while jogging,” the
i-appli music player function supports
Bluetooth A2DP and AVRCP.

1) A2DP Support

The A2DP profile is used for trans-
mitting high-quality audio data. It is
mainly used for sending and receiving
an audio stream between a terminal and
headphones [2]. Audio output from an
i-appli—whether that be audio data output
from the i-appli music player intro-
duced here or existing i-appli audio—
can now be transmitted to headphones
or other devices via A2DP. Incidental-
ly, a delay of about several hundred
milliseconds occurs in audio transmis-
sion by Bluetooth compared with wired
earphones. Thus, taking into account
music-related games and such that
require close synchronization between
sounds and user operations, each i-appli
is given the ability to enable or disable
audio output by Bluetooth.

2) AVRCP Support

The AVRCP profile is used for
transmitting signals like “play” and
“stop” to the mobile terminal from the
Bluetooth controller attached to a
device like a headset. It is frequently
used with portable music players [3].
With AVRCP support, an i-appli can be
notified of an AVRCP event transmit-
ted from a Bluetooth controller.

Installing an i-appli with AVRCP sup-
port can be used not only for operating
music playback software but also for
operating other types of functions and
services.

5. Conclusion

This article described functional
extensions for NTT DOCOMO’s win-
ter/spring 2010-2011 models. Specifical-
ly, it described the i-concier function
with extended input-support and remind
functions, the Machi-chara function
with extended speech-dialogue and net-
work-update functions, and an i-appli
that extends the music player function.
These new applications have made it
possible to achieve a service platform
that can provide a good balance between
“ease of use” and “new added value.”

Looking to the future, NTT DOCOMO
will continue to develop applications that
help users make full use of increasingly
diverse services and built-in mobile-
terminal functions and that respond
appropriately to evolving user needs.

REFERENCES

for 2009—Evolving Web Applications—,”
NTT DOCOMO Technical Journal, Vol.11,

Site: “A2DP.”

Site: “AVRCP.”