



Dhirubhai Ambani
Developer Programme



**Reliance
Infocomm**

A New Way Of Life

December 2004

R-EVOLUTION

Newsletter for the Developer Community

For Private Circulation only

Editorial

Dear Developers,

With the growing popularity of mobile phones, mobile information access and remote transactions are fast becoming commonplace. However, mobile devices have limitations on the end-user experience, with relatively small visual displays and cumbersome keypad input. Multi-modal interfaces capitalizing on the efficiency of visual displays and ease-of-speech input are expected to soon overcome these limitations of mobile devices. These interactions will enable you to speak, write, and type, as well as hear and see, using a more natural user interface than today's single mode interface.

Our lead article in this issue unravels the advantages of multi-modal applications, and the issues associated with multi-modality and the challenge to be overcome to make it a seamless experience for users.

Application features and benefits of 'Call History', particularly to the visually challenged, are highlighted in the Application of the Month section. In our continuous endeavor to equip our developers to conceive, develop and test applications, we have added one more compelling feature of uploading your executable files on the DADP website and seamlessly test the application on Reliance handset using DADP OTA tool. Please refer to New Products and Processes for more details.

In our Technology Tips section, we have provided details of phone book API to demystify its use in developing application for Reliance handsets.

As always, we look forward to receiving your valuable suggestions and useful feedback.

- S P Narayanan

Multi-modal Wireless Applications - introducing versatility in communication



Tech Tip
PhoneBook APIs



Application of the Month
Call History



**New Products &
Processes**



Interview of the Month
Daffodil Software



Quiz

Multi-modal Wireless Applications - introducing versatility in communication

Wireless users want to send text messages, browse information and services, and carry out wireless commerce transactions and access business applications, all from their mobile device by just clicking a few buttons. Consumers and business users are demanding additional flexibility in accessing their mobile applications.

Multi-modality is the next step in the evolution of data and speech services for wireless carriers. It is an exciting new technology that promises to dramatically enhance the mobile user experience. Multi-modality combines the best of both worlds speed and convenience with high utility and relevance of high-value applications that make people want to use wireless data. For mobile devices multi-modality is especially important because of small displays and missing keyboards.



Multi-modal applications combine speech, touch and graphics into a single interface, dovetailing input (touch screen, keypad and voice commands) and output (images and sounds) modes in a device to allow users to access wireless content in almost every possible way they want it. Applications are able to call upon both modes as required - and the users are able to do the same.

This brings to the fore the aspect of current mobile application delivery methods. Some of the most popular and tried methods include WAP, services such as GPRS, and speech-based services. However, these have met with only moderate success with no attractive Return-On-Investment (ROI) for operators. At the same time, analysts continue to predict more uptake rates on mobile applications as new technologies like 3G enter the marketplace. In addition, devices create significant challenges to the subscribers. Small screens, limited input capability and slow performance make dependence on these services quite irksome.

While voice interfaces overcome the form factor problems of visual interfaces, they are not practical for

communicating graphic information, long lists, or complex instructions. Such elements, common to many applications, strain one's memory and are better suited for visual interfaces. Analysts predict that in the days to come, wireless technologies will come to heavily rely on multi-modal applications, which allow users to interact with data and services visually via the device interface, by speaking, or using a combination of the two. This will put the power of choice in the hands of the user.

Multi-modality and Mobile Applications

Technology is rapidly evolving to enable wireless applications to become multi-modal - leveraging both visual and voice modes to optimize the exchange of information. This leads to a seamless integration of graphics, text and audio/touch input to dramatically deliver enhanced applications and services experience. Multi-modality really means that we interact with wireless applications using multiple modes, e.g. read, key in details, speak and listen.

Multi-modal applications offer significant value like the ability to get directions, or book travel arrangements, or access personal information all while on the go. They are easier and more intuitive to use and have the promise to increase subscriber adoption. Plus, the economic values that accrue to wireless operators, services providers, and application enablers are indeed tremendous.

Advanced speech recognition and text-to-speech on the voice side can lead to an engaging and enriching experience for SMS and MMS based services. Imagine enquiring in voice about 5-star hotels within five kilometer radius of Mumbai airport and getting the results in the form of an SMS and asking further details about one particular hotel, again using voice. Leveraging multi-modality enhances usability, flexibility, and speed as detailed or long lists of information that cannot be conveniently spoken can be visually shown.

Multi-modality offers complete control over voice and visual interaction. They include applications such as driving directions, directory assistance and electronic mail access. For example, in a navigation application, it may be easier to speak the name of the place (voice mode) than to type it, yet it may be preferable to view a map (visual mode) than to listen to directions that may involve numerous turns. The application itself may initiate the switch between the visual and voice modes or the user can do the same himself.

Multi-modality can also empower the user to simultaneously use voice and visual communication; a good example of an application using both the modes being an m-commerce application, which could facilitate transfer of funds from one account to another by populating appropriate entry boxes on the basis of utterances of the user. The confirmation of the transaction can be given in the visual mode.

Opportunities and barriers

Research says most wireless users prefer speaking information into a device rather than entering it via a keyboard. More than half are also interested in answering SMS or e-mail messages by placing a call or recording a voice reply. About two-thirds would like to read or listen to an e-mail message from their mobile devices. Developers could use this information to design multi-modal applications that match the preferences of these users.

Another example of a multi-modal application could be 'dial by voice' services that allow subscribers to make calls by simple spoken instructions. "Call Amit or dial XXXXXX no," and you are connected to your friend. Such value-added services can build loyalty for the brand while driving usage further up on the service value chain.

What users want are applications that are simple and convenient and give them more control. Voice-controlled Address Book, for example, can give phone numbers of favorite restaurants, cinema theatres etc without the need for flipping through the yellow pages. One can also create a personalized directory of friends, family, colleagues and business associates, which simplifies management of contact information.

Companies like Bell Mobility and VoiceGenie Technologies Inc, the world's leading VoiceXML Gateway solutions providers are actively working on research and development to extend VoiceGenie's existing speech recognition platform into a multi-modal platform, allowing users to access rich information, graphics and streaming video on their wireless devices using voice commands. This will make wireless applications more efficient and convenient for subscribers.

However, the current limitations of speech recognition technology are the major impediment to multi-modal offerings. This technology will have to improve and developers must create applications using technologies such as SALT (Speech Application Language Tags) before multi-modal technology sees much adoption.

India's initiatives

"Potential for multi-modal wireless applications is more

strong in India than many advanced countries, the reason being that in India the population is not fully literate. Varied environment also needs multi-modal applications, which bring in certain amount of flexibility. Reliance is planning to investigate the potential for offering multi-modal applications by 2005. We have seen demos of these services from international companies. The constituent modes of interactions namely speech, graphics should be stabilized before we offer these applications. We are currently working on location-based services," says Dr Sandeep Sibal, Technology Advisor to the Chairman's Office, Reliance Infocomm.

In India, multi-modal applications have a long way to go; most of the services being offered in this area are voice-based location services like medical shops, police stations, hospitals and taxi services. Some operators also offer cricket commentary in their voice-based services.



Dr Sandeep Sibal

The path ahead

It is clear that Wireless carriers, service and application providers, and speech technology companies believe that multi-modality is very real and crucial to the future of wireless. Analysts are predicting that by 2006, one third of all application usage will be multi-modal. However, for that to happen, multi-modal applications should be made backward compatible with 2G networks and support industry standards such as VoiceXML, WML, cHTML, and HTML. Use of automated speech recognition (ASR) and text-to-speech (TTS) technologies in mobile applications will boost unified messaging. Mobile users have neither the time nor tools for handling e-mails and faxes in their native formats. Voice technology will enable content to easily flow from one media to another. "Mobile market is rapidly growing with half a billion phones being sold worldwide annually and in the foreseeable future, multi-modal applications will have a strong value proposition. All applications will go multi-modal in future," predicts Dr Sibal.

Phone package of Reliance APIs

Following the 'media package' that was featured in the last issue of R-evolution, this issue of R-evolution brings you the *phone* package of Reliance APIs with illustrations.

Phone package is the second most sought after package after the 'media package'. This package is mainly used to control the native functionalities and features of the handset and includes phonebook, alarm, scheduler, etc. The package has four classes namely, *Alarm*, *PhoneBook*, *Scheduler* and *Util*. This article would be discussing the *PhoneBook* class. Other classes will be discussed in the forthcoming issues of R-evolution.

PhoneBook class has a set of methods for accessing and manipulating the device's native phone book. Class has eight methods namely, *addDataEntry*, *addNameEntry*, *deleteDataEntry*, *deleteNameEntry*, *getDataEntry*, *getEntries*, *getName* and *getSupportedFields*. Along with these methods, Class also has eight constants namely, ADDRESS, DATA_PHONE, EMAIL, HOME_PHONE, MOBILE_PHONE, OFFICE_PHONE, PAGER_PHONE and PRIMARY_NUMBER. The method *addDataEntry* and *deleteDataEntry* are used to add and delete data related to a name entry in the device's phone book. The method *getDataEntry* retrieves data related to a name entry in the device's phone book. It returns a string containing the requested data or 'null' if an entry for the given data type does not exist. For phones that have fixed data structure, if an entry for the given data type does not exist, it returns an empty string. Method *addNameEntry* adds a new name entry to the device's phone book. If an entry with the same name already exists, then a new entry will be added, provided the device supports multiple entries with the same name. If the device does not support multiple entries with the same name, the method simply returns. The return value from the call will be the ID where the name is now available. Method *deleteNameEntry* will delete a name entry from the device's phone book. All data related to this name entry will also be deleted. Method *getEntries* is used for looking up entries in the device's phone book. It allows receiving either a list of all entries or a sub-list containing only entries that have names that begin with a certain prefix. The method returns an *int* array containing all the relevant entry IDs. The method *getName* returns the name of a phone book entry corresponding to a particular ID. The method *getSupportedFields* is used to query the structure of the device's phone book. A single name entry can have any number of phones related to it. It returns an integer array containing an entry for each phone type. Entries in the array will be one of HOME_PHONE, OFFICE_PHONE, PAGER_PHONE, MOBILE_PHONE, DATA_PHONE, PRIMARY_NUMBER, ADDRESS or EMAIL. If the phone book only supports simple name-phone entries, this method will only return PRIMARY_NUMBER.

The sample code for the implementation of *PhoneBook* class is given in the appended list. The method *getNameDetails* returns a two dimensional array of all the details of all names from the phonebook of the handset that starts with the given string.

For further details on the Phone package of Reliance APIs, please download the complete documents of Reliance APIs from the DADP website www.dadp.com

```
private void getNameDetails(String fname)
{
    if(fname.equals(""))
    {
        showAlert("Error","Invalid Entry.");
        return;
    }

    int Entries[] = PhoneBook.getEntries(fname);
    int totEntries = Entries.length;

    if(totEntries == 0)
    {
        showAlert("Error","No entries found.");
        return;
    }

    int supportedField[] = PhoneBook.getSupportedFields();
    int fieldLength = supportedField.length;

    String nameDetails[][] = new String[totEntries][fieldLength + 1];

    for(int i=0; i<totEntries; i++)
    {
        nameDetails[i][0] = PhoneBook.getName(Entries[i]);

        for (int j=0; j<fieldLength; j++)
            nameDetails[i][j] = PhoneBook.getDataEntry(Entries[i], supportedField[j]);
    }

    for(int i=0; i<totEntries; i++)
        for(int j=0; j<=fieldLength; j++)
            if (nameDetails[i][j].equals(null))
                nameDetails[i][j] = "No present";
    }
}
```

Creating (Call) History

Jayadev Gopalakrishnan and Anupam Varghese of Tinfo Mobile, Trivandrum, conceptualized the 'Call History' application which won a special prize at the first ever Mobile Applications Contest in India, organized by DADP.

The DADP team took the prize winning application forward by optimizing it for Reliance environment and thereby enabling the goal of fulfilling a noble social cause. Targeted at the visually challenged, 'Call History' reads out the missed, dialed and received calls list on Reliance phones and features an innovative J2ME based Number-to-Speech engine.

Not only does it enable a visually challenged person to access information on missed, dialed and received calls but it also provides easy navigation through the menus on hearing the voice prompts by pressing the appropriate number key. The user can also selectively 'listen' to any of these call numbers and choose to either call back or delete the number from the respective lists. The application interface has been designed to be simple and intuitive so that even a first time user finds it very easy to handle. The user can navigate and select from the menu items by listening to the voice prompts, namely Press 1 for Missed Calls, Press 2 for Received Calls, Press 3 for Dialed Calls and press 4 for Exit.

Hurdles and breakthroughs

Developing such an application involved overcoming some technical snags. First, it was observed that there was no provision to access sensitive data (like SMS and Missed Calls list) from cell phones using Java. Secondly, most devices did not allow applications to output wave audio. Thirdly, the application necessitated the use of a TTS (Text-To-Speech) conversion engine on a mobile device. Rummaging through the Reliance APIs (Application Programming Interfaces) and certain other references provided by the DADP team, the Tinfo team located certain restricted APIs that allowed access to the Missed Calls list and SMS. Further inspection revealed that APIs could also be used to play audio. However, the biggest challenge for Tinfo was solving the TTS functionality.

After much thought, it was decided to design an application with limited TTS capabilities. It was realized that most of the data would be either numbers or instructions.

Ultimately, the need was zeroed down into making a simple speech engine that took numbers and a limited set of instructions as input and gave audio as output.

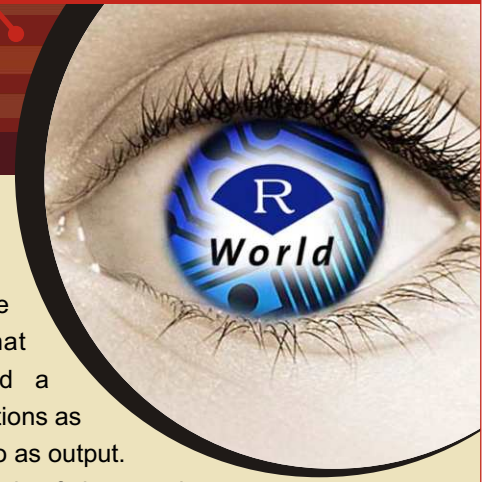
This marked the birth of the epoch making 'Call History' application. The application underwent intensive testing, including live trials by the visually challenged, before it was finally cleared for deployment on Reliance IndiaMobile phones.

Access Mechanism

To launch the application, the user needs to dial *444 from his RIM handset to automatically download the application on the handset. Currently, only MA2 compatible handsets (i.e. handsets that support voice ringtones) such as Samsung Rainbow, Samsung XBS, Samsung RCP, LG 7130 and LG 7230 can be used to launch the application. If the handset is not compatible, an SMS is sent out, informing the user of the same.

"Call History enables us to keep track of our calls like any sighted person. It also helps us in business as we can be more productive and forthwith in contacting our clients. We hope that this is just the beginning and future will see more such applications being developed for the visually challenged," says Ketan Kothari, Deputy Director, National Association for the Blind.

The 'Call History' application has been showcased as a pioneering effort of Reliance Infocomm in fulfilling social obligation towards the visually challenged. Though the application is primarily meant for the benefit of the visually challenged, it is expected to serve as a high utility tool for anyone using a Reliance mobile phone. "The Call History application is a great example of Reliance's spirit of empowering people and promoting innovation. It shows how technology can be harnessed to touch the lives of so many people," concludes Amit Chandra, Product Manager responsible for the application.



Amit Chandra

Direct upload to Reliance environment



The DADP team has introduced a new feature the 'Direct Upload facility to Reliance Environment' - of the OTA (Over The Air) testing tool. In August 2004, the DADP developer community was introduced to a new testing tool to test their applications (August 2004 issue of R-evolution under New Products and Processes). The developers were asked to send the client side executables to the DADP Team, which was then uploaded on the Reliance environment. These applications were downloaded using the DADP OTA Tool on the handsets and tested. This entire process reduced the time for the developers but there was still a dependency on the DADP team to upload the application on the Reliance environment.

Now, the 'Direct Upload facility to Reliance Environment' feature eliminates this dependency. Developers can themselves upload the applications directly on the Reliance Environment and test them. The tool can be used for testing both, client only as well as client-server applications. In case of client-server applications server side components have to be deployed on either the Reliance platform or on the public domain. The client must then use DADPRequest and DADPResponse classes to implement the same (refer to article on DADP OTA Tool on www.dadp.com). Developers need to get their RIM handsets provisioned through a simple process. Only handsets provisioned by the DADP team will have access to the DADP OTA Tool.

Here is a step-by-step procedure to use the 'Direct Upload to Reliance Environment' feature of the DADP OTA Tool:

- 1) Send a request to dadp.query@relianceinfo.com to get your RIM provisioned for the testing tool. The request should consist of the RIM number and the phone make and model (example LGRD 2030) of the handset. Along with this, a brief description of the application developed or under development is to be sent for activating the handset for DADP OTA Tool.
- 2) Developers will then have to do a registration on the DADP website.
- 3) The user can give any username he wants, but it is advisable to have a small user name, preferably 3-5 characters, as the same username will be used for login into the handset for testing. Choosing a long username will be inconvenient while entering the same on the handset for login.
- 4) The password of the account will be sent as an SMS to the developer on the RIM, specified during registration on the DADP OTA Tool website. This password can later be changed from the site itself.
- 5) Prepare the JAR and JAD file for upload. Remove the entire Reliance package from the JAR file, except the com.reliance.ui package (if you are using this package). All other packages of the Reliance APIs are burnt on the handset and should not be present in the application JAR. Update the size of JAR in the JAD file.
- 6) If the application is a Client Server Application, then update the User defined key: `_rapurl` to the URL of the server where the application is hosted. If the server side component of the application is hosted on Reliance environment, then update `_rapurl` with the Reliance server URL.
- 7) Use the username and password to login to the DADP OTA Tool section of DADP website to upload the application JAD and JAR files.
- 8) Launch the R World on the handset and go to the following location. R World -> More Services -> Tools -> DADP -> DADP OTA. On launching the DADP OTA application, it will ask you the login username and password. Use the same username and password that was used to upload the application on the website.
- 9) Use Download option in the application to download your application on the handset and launch it. Once the application is running, it can be tested for its functionalities.

It is hoped that the tool will be of great use to developers in testing their applications and developing world-class mobile applications. Happy testing.

"With technological growth, users expect the same functionalities from mobile phones as they do from their desktops!"

Brothers Yogesh and Puneet Agarwal started Daffodil Software Ltd in 1999 with the financial backing of its parent concern, the US\$ 63 million DARCL (Delhi Assam Roadways Corporation Limited), which is India's second largest privately held transport company. Headquartered in California, Daffodil has state-of-the-art software development facilities located at Gurgaon and Hisar in Haryana.

Daffodil's product set consists of Daffodil DB (a J2EE-certified, SQL-99 and JDBC 3.0 standards compliant Java database), One\$DB (a free version of Daffodil DB) and Daffodil Replicator (a bi-directional data synchronization tool). By open sourcing its product, Daffodil Software has shown its commitment towards the Open Source Community. "To offer cost-effective and high-quality database management tools and solutions that evolve with corresponding technology trends like mobile applications is our mission," say the brothers exuding pride.

In a free wheeling interview with R-evolution, Puneet Agarwal, CTO of Daffodil Software Ltd, spoke about Application development work and their applications. The following are excerpts from the conversation:

Please tell us about your experience as a Developer working for Reliance. What were your expectations?

Working for Reliance has been a great experience. I would rate this as a great opportunity for me. We received prompt support and were communicated the minutest of information with complete details and sometimes, even by phone conference. The team was very cooperative and enthusiastically cleared my technical and non-technical doubts.

What prompted you to develop the AutoCab fare application?

I see that auto and cab owners often dupe people who are new to the city. This has also happened to me while traveling to other cities. When I had the option to develop an application for Reliance, I was delighted to choose the AutoCab fare application.

What are the unique features/functionalities of this application?

We tried to make this application very flexible and robust for end-users as well as for the administrator of the application. RIM users can get all necessary information like distance of the location where one wants to go, as also the day and night fares. The user also gets suitable messages when autos and cabs do not ply from the location he/she chooses.

Through the 'ticker' at the client end, Reliance can even insert related advertisements of tourist and taxi organizations. The administrator can directly work on live data and all changes will be reflected to the end-user when the administrator presses the 'activate' button.

What is the future / scope of the application?

Currently, this application is active in five major cities in India and we expect it to be scaled up to ten shortly. This can be further expanded for other cities, by providing fare data of those cities.



Puneet Agarwal

What other applications have you developed or are developing for Reliance?

We are in the process of developing a software, which can be used to track vehicles all over the country. This would be a useful application for transport companies who would want to know the location of their vehicles at any point in time. The large Reliance network will be the soul of this application. This application needs supporting APIs from the RAP Library.

Any interesting or memorable event that you would like to share about your work with the DADP team?

Although every moment of working with the DADP team has been memorable, I will never forget the moment when I actually saw the AutoCab application running on my handset for the first time. It will always make me feel good that this application is being used by millions of people.

Any comments on the mobile applications scenario in India today and the future trends?

When mobile technology was first introduced, mobile devices were restricted to voice transfer. Recent technological growth has enabled these tiny handsets to transfer data too. Now, users expect the same functionalities from mobile phones as they do from their desktops! More importantly, this has become feasible! However, we have a lot of work cut out, if we want to develop and market applications that are accepted the world over. Having said that, there is a big revolution knocking at our doors and Reliance has identified the knock.

Dear Sir,

Thank you for the newsletter. The media package described is proprietary Reliance APIs. Since MIDP 1.0 does not support playing of audio and video files, unlike MIDP 2.0, can you let us know by when Reliance handsets would support MIDP 2.0, so that we can develop multimedia rich applications?

Rahul (Wireless Application Developer)

Editor's Response: Thank you for your feedback. Reliance specifications for MIDP 2.0 are being finalized and handsets with MIDP 2.0 support will be available in the near future.

Dear Sir,

I want to write a series of articles for developers who are interested in developing wireless applications. The articles will cover both Reliance and Global context.

Kunal Deo

Editor's Response: Thank you for your email. Please do send us your articles. The Editorial Committee would review and publish relevant articles in the Newsletter.

- Which of the following classes allow synchronized read operations by multiple threads?
 - Hashtable
 - TreeSet
 - HashMap
 - HashSet
- Which of the following classes allow unsynchronized read operations by multiple threads?
 - Vector
 - Hashtable
 - TreeMap
 - Both a and b
- Which of the following are true statements?
 - The relationship between a class and its superclass is an example of a 'has-a' relationship.
 - The relationship between a class and an object referenced by a field within the class is an example of a 'has-a' relationship.
 - The relationship between a class and an object referenced by a field within the class is an example of an 'is-a' relationship.
 - None of these
- Which interface of the java.util package offers the specific behavior of 'Entries organized as key/value pairs' and 'Duplicate entries replaces old entries'?
 - List
 - Map
 - Set
 - None of the above
- Which global giant is responsible for commercializing CDMA technology?
 - Quark Systems
 - Qualcomm
 - AT & T
 - None of the above

Quiz



Monthly Poser

BREW, an application platform, has similarity in coding with which other language?

- SQL
- C Programming
- Oracle
- None of the above

Answers : 1. A) 2. C) 3. B) 4. B) 5. B)

Answer to last month's poser: Personal Java

Winner of the last month's poser (through a lucky draw is): Ms. Sailaja Parthasarthy, IBM Rational Software, Bangalore.
Winner will receive her prize by courier.

Answers to the Monthly Poser should be sent to dadp.newsletter@relianceinfo.com mentioning Monthly Poser - December 2004 as the subject. The winner will be decided on the basis of a lucky draw and walk away with a prize from DADP.

You can contribute ideas and information to R-evolution at the following e-mail address: dadp.newsletter@relianceinfo.com.

Please note that contributions may be edited for clarity, style or length.

Editorial Team : S P Narayanan, Saurabh Chakrabarti, Amit Behl, Amit Chandra, Suresh Dabbara, Nitin Saxena, Vikram Vishwanath, Lakshman S Aiyar, Vijayam Raghunathan

Design Team: Makrand Pange, Jaydeep Gholap

Published by the DADP and Corporate Communications Teams, Reliance Infocomm, DAKC, Navi Mumbai