DataPhone: An Intelligent Phone for Data Conferencing

Lap Chung Lam
Rether Networks Inc.
lclam@cs.sunysb.edu

Peitao Deng
Stony Brook University
pdeng@cs.sunysb.edu

Tzi-cker Chiueh
Rether Networks Inc.
chiueh@cs.sunysb.edu

Abstract— **Data conferencing is one of the more commercially successful examples among conferencing applications, because of its effectiveness and relatively modest resource requirements. Existing data conferencing applications are built almost exclusively on IP-based packet switched networks. While they are very effective in improving office communications, their use in the households is still quite limited because data conferencing end-points are mostly personal computers and are thus not particularly friendly to a majority of the population in the world. In contrast, more than 95% of the population in the United States are quite comfortable with conventional telephones. This paper describes the design, implementation, and evaluation of an information appliance, called DataPhone that can be used both as a regular phone and a data conferencing device. A DataPhone consists of a touch sensitive screen, a SVD modem and a microcontroller-based single board computer. DataPhone allows two parties to have a regular voice communications channel, as well as a shared whiteboard for sketching and for document sharing. Typical applications of DataPhone include annotating an engineering design while discussing its technical details, drawing a map to a person's house for a friend, and scribbling random information down for future reference. One of the key design decisions of DataPhone is that it runs directly on PSTN, and does not required any additional infrastructural support, such as ISP subscription, domain name service, routing gateway, etc., thus avoiding their associated configuration headaches. We have successfully built a fully functional DataPhone prototype and evaluated its usefulness through end user testing. Initial responses are both favorable and encouraging.**

I. Introduction

Despite the development of many intelligent network (IN) features over the years, the basic functionality of conventional phones remains largely the same since its inception: voice-only communication. The original motivation of this project was to explore what fundamentally new functionalities can be added to standard phones such that (1) no infrastructural modification is required and (2) the "user interface" of the phone does not need to be changed. The second requirement is important because it is estimated that while less than 30% of the U.S. population are comfortable with computers, more than 95% of them know how to operate phones without any difficulties. Therefore, preserving the interaction model of conventional phones while developing new functionalities is essential to minimize resistance to widest possible adoption. The result of our investigation is an intelligent appliance called DataPhone that can serve as a regular phone as well as a data conferencing device over the standard PSTN.

A data conferencing system typically supports a voice communication channel and a whiteboard for document sharing and/or annotation. Compared with video conferencing systems, data conferencing systems require much less resource and yet are almost equally effective. However, most if not all existing data conferencing systems are built as applications running on personal computers over IP networks. As a result, while they prove to be quite useful in improving office communication effectiveness, their usage in the households is still quite limited. One important reason is that the majority of the population, even in the U.S., are not sufficiently familiar with computers to feel comfortable with data conferencing applications. Another reason is that IP connectivity from households is still largely based on dial-up modems, and requires a much longer time to set up than conventional phone calls. These two reasons together make typical grandmas unlikely users of data conferencing systems.

The goal of the DataPhone project is to show that it is possible to build data conferencing service into standard phones such that existing phone users can use it as an extension to the standard voice service. Therefore, all existing phone users are potentially the users of DataPhone. DataPhone provides both a voice communication channel and a data communication channel over a single PSTN connection. The data communication channel supports whiteboard annotation, file transferring, and shared document viewing. DataPhone allows two mechanical engineers to discuss the details of an engineering design diagram over the phone, a lawyer to explain the subtleties of a contract to her client, a person to draw a map for his friend how to get to his house, two hearing- or speaking-impaired people to communicate with each other, etc.

A DataPhone consists of a touch-sensitive screen, a Simultaneous Voice and Data (SVD) modem, and a microcontroller-based single board computer. The SVD modem makes it possible to support a voice communication channel and a data communication channel on a single regular phone line. The microcontroller is responsible for managing annotation, communication, and controlling the SVD modem and display. The specific design goals of DataPhone are

1) DataPhone should be able to work seamlessly with regular phones. That is, when a DataPhone calls a regular phone or receives a call from a regular phone, it should be able to establish a normal voice-only communication with the regular phone. On the other hand, if the remote party is also a DataPhone, it should support both voice and data communications.
2) No additional infrastructural support in any form, including Internet service subscription or extra gateways or servers, is required. Users should be able to plug a DataPhone into a standard phone jack and start using it, just like any other normal phone.
3) A shared whiteboard for digital pen drawing and writing should be provided, and these digital pen annotations can be saved for later reviewing or printing.

4) DataPhone should support file transfer, synchronized document viewing, collaborative annotations on shared documents, and storing annotations with documents. This feature allows a lawyer to go over a legal document item by item with her client, and the client to sign it at the end, both over the phone.

Video conferencing systems [7] have been around for a long time, but they are in general expensive equipments that are both expensive to acquire and operate, and thus are beyond the reach of most consumers for personal use. A special instance of video conferencing systems is videophone [2], which had meteorically emerged in the late 90's only to fail miserably in the market space because either it required expensive ISDN connectivity or the resulting quality over PSTN lines was too poor to be acceptable. Besides its economic and engineering feasibility, there is an issue of how desirable it is socially to use videophones in informal communications because it forces participants to be dressed up and to pay undivided attention at all time. Tegacky [6] is a product of Toshiba product that supports touch-pen entry for email communication over the PHS network, in addition to normal voice communication. It allows users to compose and send/receive freehand-written messages, and comes with a large, easy-to-view LCD display. Touchphone [5] is a Linux-based phone from Sorgenti in Italy that supports a similar functionality to DataPhone. However, from the publicly available documentation, it appeared to be based on IP, and therefore was not compatible with regular phones. There are also many IP phones on the market [3], [4] that support various conferencing functionalities that are similar to DataPhone. We chose not to take the IP-based approach for two reasons. First, the quality of service on public Internet is still largely unresolved, especially for consumer users (as opposed to enterprise users). Second, IP connectivity typically requires special set-up, such as DNS, router, DHCP, etc, and thus is generally considered unfriendly to the mass. In particular, except always-on broadband access service, it still takes much longer to establish IP connectivity to a service provider than a typical phone call. In contrast, DataPhone works exactly like a normal phone, and thus requires no special skills and server equipments. The only infrastructural requirement for DataPhone is a regular telephone line, which is ubiquitous.

The rest of this paper is organized as follows. Section 2 outlines the system architecture of DataPhone. Section 3 describes the implementation details of the current DataPhone prototype. Section 4 presents the results of a user study and their usage experiences. Section 5 concludes this paper with a summary of DataPhone's features and a brief outline for future work.

II. DataPhone Architecture

The system architecture of DataPhone consists of three layers as shown in Figure 1. The GUI layer provides users with menus and buttons to control the DataPhone, and a display area for writing, sketching, and displaying document images. The GUI layer is composed of two modules, one of which supports digital pen sketching while the other is for document image display and management. The GUI layer captures local events, packages them into packets, and sends them to the remote end through the Communication layer. Similarly, when the GUI layer receives remote events from the Communication layer, it interprets them properly and performs corresponding operations against the screen. Possible events include pen sketch, button press, and window scrolling.

The Communication layer provides basic communications service, file transfer service, as well as a high-level floor control mechanism to synchronize the operations against the screen. Since all operations against the screen from either side of a DataPhone connection need to be synchronized, the GUI layer is responsible for recording such events as choosing a menu item, clicking a button, or drawing a line, and packages them into packets. The Communication Layer delivers these data packets the GUI Layer generates to the other end. The synchronization is based on a simple token protocol: A communicating party controls the floor only when she successfully grabs the token. Finally, the Communication layer also offers a file transfer service that supports compression/decompression, and uses a simple stop and send protocol for error and flow control.

The Modem layer handles connection setup, teardown, and low-level byte stream transfer. There are three possible types of connections, DataPhone to DataPhone, DataPhone to normal phone, and normal phone to DataPhone. Upon initiating or receiving a call, DataPhone employs a simple handshaking protocol to detect the type of remote device automatically, and establishes a proper type of connection accordingly. This layer isolates the rest of the DataPhone system from the peculiarities of specific modem devices.

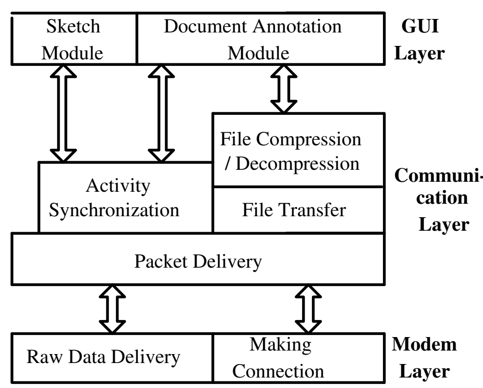

Fig. 1. The DataPhone system architecture consists of three layers, GUI Layer, Communication Layer, and Modem Layer. The GUI layer provides users with menus and buttons to control the DataPhone, and a display area for writing, sketching, and displaying document images. The Communication layer provides basic communications service, file transfer service, as well as a high-level floor control mechanism to synchronize the operations against the screen. The Modem Layer is responsible for connection setup, teardown, and device detection.

III. Prototype Implementation

This section presents the implementation details of the current DataPhone prototype, whose design is based on the system architecture outlined in the previous section.

A. Hardware Requirement

Ideally a DataPhone should look like a regular telephone with an extra touch sensitive screen. However, to reduce the prototyping effort, we did not design the DataPhone hardware from scratch. Instead, we built the current DataPhone prototype out of a PDA and an external SVD modem.

The PDA is used to host the main control program, which manages annotation and document data, supports data and file exchanges, and provides a graphical user interface. We chose a low-end PDA, PalmPilot professional, in this prototype to demonstrate that DataPhone's computation requirement is quite modest. PalmPilot comes with an OS that supports basic GUI programming, and serial port communication management. The DataPhone control program controls the operation of the SVD modem using modem AT commands [8] through the serial communication port API [9].

SVD modem comes in two varieties [1], Analog Simultaneous Voice and Data (ASVD) or AudioSpan SVD modem, and Digital Simultaneous Voice and Data (DSVD) modem. Both ASVD and DSVD provide simultaneous full-duplex transmissions of high quality voice and data over worldwide telephone networks. ASVD and DSVD implementations differ in the manner that voice and digital data are combined. DSVD uses a technology known as DigiTalk to compress the human voice into digital data, and the compressed digital voice data is combined with other digital data together into data streams for transmission. ASVD uses phase encoding and decoding to transfer the voice and data. Although DSVD can provide higher data bandwidth, ASVD modem was chosen in the DataPhone prototype because of its lower cost and its ability to transmit music in analog form.

A key advantage of SVD modem is that transmission of voice is completely transparent to the programmers because voice is transmitted directly by the modem hardware, either digitally (DSVD) or in analog form (ASVD). As a result, the CPU overhead related to voice transmission and reception is completely eliminated. This reduces the hardware cost of the controller and simplifies the control program. ASVD modem operates in two modes, voice mode and data mode. ASVD modem works like a normal telephone in the voice mode, and it can transmit voice and data simultaneously over a single phone line in the data mode. Due to this two-mode operation, DataPhone is able to work seamlessly with normal telephones.

The resulting DataPhone prototype, while not as integrated as a commercial product should be, is a very close approximation, in terms of functionality and performance, to its commercial version that integrates all the components into a single box.

B. GUI Layer

The GUI Layer provides the users with a simple graphical user interface that consists of three major windows. The first window (left-hand side of Figure 2) is the connection window that is designed to look like a cellular phone interface, and enables the users to make phone calls exactly like a normal phone. The second window is the sketch window (right-hand side of Figure 2) against which users can write or draw. The third window is the document annotation window (Figure 3) for displaying and/or annotating document images. The GUI Layer also monitors all events such as menu events, button events, and drawing events in the main event loops. Each time when it captures an event, it constructs a data packet describing the event and asks the Communication Layer to deliver the packet to the remote DataPhone. When it receives a packet from the remote DataPhone, it interprets the packet, and performs the corresponding operation. For example, after a remote DataPhone draws a line, a DataPhone's local GUI Layer will receive a packet describing the line, and it has to draw the exact same line on its screen. Consequently, both DataPhones display exactly the same thing on their screens.

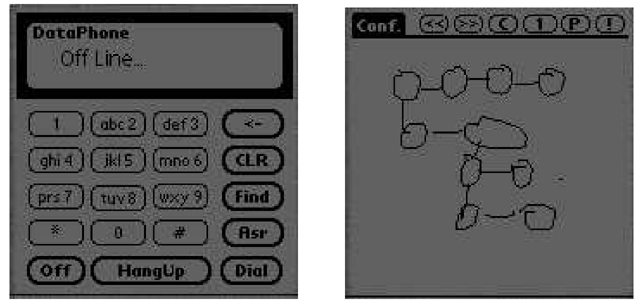

Fig. 2. The left-hand side is the Connection Window, which is designed like a cellular phone interface and enables users to enter a phone number, dial, answer, or hang up a phone call. The right-hand side is the Sketch Window, which is a whiteboard for drawing and writing, and provides the functionalities for user to create a new page, go to another pager, clear a whole page, choose different pen patterns, and erase drawings.

1) Sketch Management: The sketch window (right-hand side of Figure 2) is the primary window for drawing and writing. It has six major buttons for page navigation and a hidden menu for more selections, such as saving a page, or initiating a file transfer. To make the best out of the limited screen size, all the buttons related to drawing are aligned with the window's title, and they are made as small as possible so that more useful buttons can be displayed. This design gives maximal space for drawing and document display. With the control buttons, a user can choose to draw or erase lines, create a new page, clear a whole page, or go to a different page. Since only two colors, black and white, are used for drawing, different pen patterns such as fine pen and median pen are implemented to enhance the drawing quality.

2) Document Image Display and Annotation: In this window both users can view a document and annotate on the document simultaneously. Figure 3 demonstrates how a document is displayed and annotated. The left-hand side of Figure 3 displays a portion of a document with annotations on it, while the right-hand side shows how the original document image is not affected by an annotation erasure operation. The screen size is only 160x160, but DataPhone supports the images with arbitrary height and width. Therefore, at any time only some portion of a document can be viewed. To make it easier for users to navigate within a document, the document annotation window also includes two scrollbars and five buttons for users to move the document image up, down, left of right in screen unit. The users can choose to save the annotation with the document together when a document annotation session ends. During a document sharing and annotation session, a floor control mechanism, to be described later, is used to synchronize the contents of the screens at two ends of a DataPhone connection.

Annotation is not directly written into the underlying document image. A document image file will not be modified unless the users want to save the annotations together with the document. To display a document image and its annotations simultaneously on the same screen, DataPhone maintains an internal annotation buffer that has the same size as the document image. Originally, the annotation buffer is initialized with the screen color. Figure 4A shows how DataPhone displays

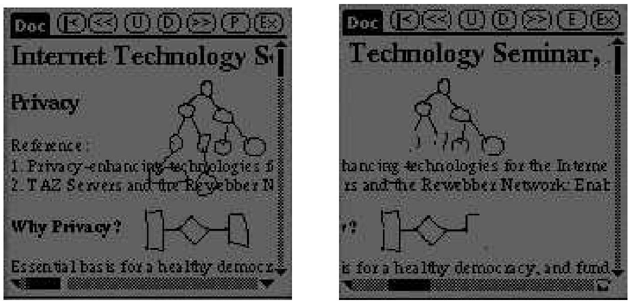

Fig. 3. The left-hand side shows a Document Annotation window that displays a portion of a document with some annotations on it, while the right-hand side shows how the underlying document image is not affected by an annotation erasure operation. The document annotation window provides different navigating buttons to facilitate navigation within a document, as well as the ability to annotate a shared document.

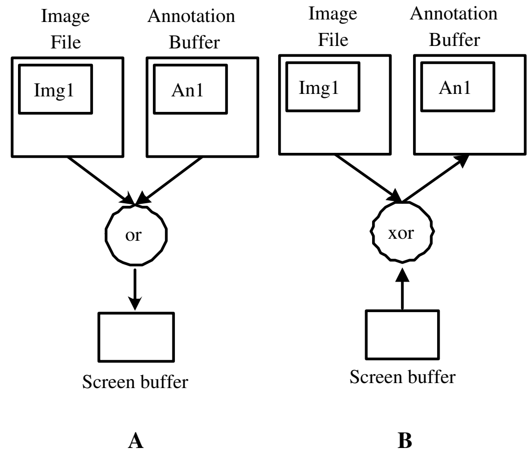

Fig. 4. Algorithm for Document Annotation. Img1 is a portion of the document image, and An1 is a portion of the annotation buffer. The annotation buffer is of the same size as the document image. Img1 and An1 are also of the same size. An OR operation combines Img1 and An1, and the result is displayed on the screen (screen buffer). The XOR operation extracts the annotations and save them back to the annotation buffer. The OR and XOR operations allow users to annotate a document image without modifying its underling file.

annotations and document image on the same screen. First DataPhone ORs Img1 (a small portion of the document image) with An1 (a small portion of the annotation buffer), and then copies the result to the screen buffer. Img1 and An1 are of the same size. The OR operation combines a document image and its annotations together. If the users want to go to the other portion of the document image, the annotation on the screen has to be saved back to the annotation buffer first. Figure 4B shows how DataPhone saves back the new annotation to the annotation buffer. First, DataPhone XORs Img1 with the screen buffer, and then copies the result to An1. The XOR operation extracts the annotations from the screen buffer, which contains both annotations and a portion of the document image. If one of the users erases some annotations on the screen, the part of the document image under the erased area is also erased. To create the illusion that the image on the screen is not affected, DataPhone performs the procedure of Figure 4B first to save the annotations, and then it performs the procedure of Figure 4A to refresh the screen. If a user decides to save annotations with a document image, DataPhone ORs the document image and the annotation buffer, and stores the result back to the document image file.

C. Modem Layer

The main task of the Modem layer is to set up connections and to hide the modem hardware characteristics from the rest of DataPhone. Whether DataPhone behaves like a regular or a data conferencing phone depends on the remote device; therefore, DataPhone needs to distinguish the following three scenarios: DataPhone to DataPhone, DataPhone to regular phone, and regular phone to DataPhone. This means that DataPhone must be able to detect what type of device it is talking to during the connection set-up time. To address this problem, we designed a novel handshaking protocol that exploits particular characteristics of SVD modems.

Modems can be instructed to generate and detect DTMF (Dual Tone Multiplexed Frequency) tone signals (1-9, a-b, *, and #). We utilize this feature to implement a DataPhone Connection Handshaking Protocol (DPCHP), whose control flow is described in Figure 5. When a DataPhone originates a call, it does not know anything about the remote side; therefore, it relies on the called DataPhone to initiate a SVD connection. After a DataPhone dials a number, it goes into the Voice Communication Phase (or Online Voice Command Mode) immediately and waits for an incoming SVD connection request. If the remote device is a regular phone, the calling DataPhone will remain in the Voice Communication Phase. Although it continues to wait for a SVD connection, this waiting will not affect the normal conversation between a DataPhone and a regular phone. If the called device is also a DataPhone, the called DataPhone will generate the **A# DTMF* sequence immediately after it answers the call. The calling DataPhone, after detecting the **A# DTMF* sequence, returns the **A# DTMF* sequence to the called DataPhone. After the called DataPhone detects the **A# DTMF* sequence from the calling DataPhone, it issues an ATD command to the modem to initiate a SVD connection. As soon as the calling DataPhone receives a RING message from the called DataPhone, it issues an ATA command to the modem to accept the SVD connection. At this moment, both DataPhones become silent for about 20 seconds to fully complete the SVD connection. Once the SVD connection is successfully established, both DataPhones enter the SVD Communication Phase, and they can exchange data and voice simultaneously from this point on.

There are two small problems associated with this handshaking protocol. First, when a DataPhone answers a regular phone, the calling party will hear three short sounds, which correspond to the **A# DTMF* tones generated by the DataPhone to request a SVD connection. The second problem is that it takes about 20 seconds for two DataPhones to complete a SVD connection. This is the time required by SVD modems to execute commands and negotiate communication parameters.

D. Floor Control

Sharing a screen using DataPhone requires a floor control mechanism to synchronize the actions of the communicating parties. More concretely, there are two types of operations

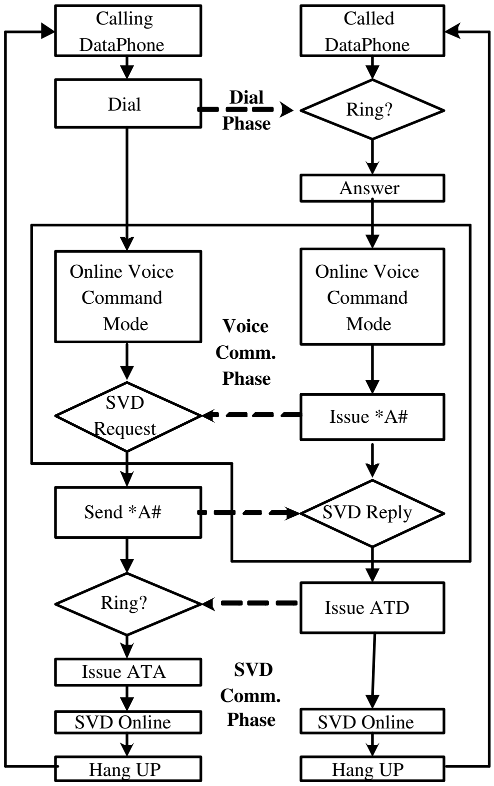

Fig. 5. Device detection and connection algorithm. After a DataPhone initiates or answers a phone call, it immediately enters the Voice Communication Phase and acts like a normal telephone. When staying in this phase, a called DataPhone sends the **A# TDMF* sequence to the remote device to detect if it is also a DataPhone. After receiving the **A# TDMF* sequence from the calling DataPhone, the called DataPhone initiates a SVD connection, and then both DataPhones enter the SVD Communication Phase that supports both data and voice communications.

against the screen that need to be synchronized, commands and drawing. Without such floor control, it is difficult to maintain consistency between the screens of the participants. We have developed a method called Token-based Step Locking that synchronizes the commands from the two communicating parties, but allows them to draw simultaneously if necessary.

After a DataPhone connection is established, the caller creates a control token in the Communication Layer. Every time a DataPhone receives data, it also receives the token. Only when the DataPhone has the token it can send data with the token. This means that each user can use the token to perform only one operation. If a DataPhone needs to perform some operations but it does not have the token, it needs to request the token from the other party first. If the other party is idle, it sends back the token immediately; otherwise, the token will be sent after it finishes the current operation. If the token is lost due to communication error, the caller creates a new token when it does not have the token but it receives a token request. The drawback of this synchronization mechanism is that only one person can draw at a time. To improve the response time, we allow users to draw simultaneously. The token-based step locking algorithm introduces a one-entry data queue to hold a single data packet that cannot be sent because the token currently resides at the remote end. When the Communication layer receives the token and the data queue has data, it sends the data immediately and empties the queue. There is a global lock in the GUI layer. A user can only issue commands when the lock is free; otherwise, the GUI layer will not respond to the user commands. However, even if the lock is not free, a user can still draw if the queue is free, and the drawing data will be stored in the queue until the token is received. To free the lock, the GUI Layer consults with the Communication layer periodically. It can free the lock if the Communication layer has the token.

E. File Transfer

Before a file can be sent to the remote DataPhone, the local DataPhone first sends a *FileTransferOpen* packet with the file name and file size to the remote DataPhone. After receiving the packet the remote DataPhone first examines the file system to see if it has the required disk space. If the remote DataPhone has the required space, it sends back a *FileTransferOk* packet. When the FileTransferOk packet is received, the file transfer begins; otherwise, the file transfer is aborted.

Each file data packet includes a sequence number. After the local DataPhone sends out a file data packet, it simply stops and waits for the ack with the same sequence number from the remote DataPhone. If the local DataPhone does not receive the ack within 0.5 seconds, it will send the same packet out again. If the remote DataPhone receives a duplicated packet, it just ignores it and sends back an ack. The purpose of this simple send and stop mechanism is to prevent the sending side from overflowing the receiving side's receiving buffer. The reliability of bit transfer totally depends on the error correction logic of serial port communications.

In the worse case, the throughput of the ASVD modem in the DataPhone prototype is only 4800bps. It thus takes about 60 seconds to transfer a 30KByte file. Therefore, a file has to be compressed before it is transferred. There are two requirements for file compression. First, the compression algorithm cannot be too computationally expensive since most PDAs have limited computation power. Second, the algorithm should support on-the-fly compression and decompression. To satisfy these two requirements, we chose Run Length Encoding (RLE), which compresses a bitmap image of recurring pixels as a single pixel value and a repetition count. This algorithm is very effective since the ImageViewer format in our implementation has only a single bit color bitmap. As a result of the RLE compression algorithm, it now takes fewer than 20 seconds to transfer the same 30KByte file.

F. Universal Virtual Image Printer

To support document sharing, DataPhone needs to be able to import documents from standard PCs. Although file transfer between a DataPhone and a PC can be easily done through PalmPilot's HotSync application, it is impossible for DataPhone to interpret all document formats available on PCs. To solve this problem, we developed a universal virtual image printer that converts different document file formats to a uniform format by capturing the images that are about to be sent to a physical printer when documents are printed. This format, called the ImageViewer format, is a simple bitmap image format, and greatly simplifies the implementation complexity of combining document images and digital pen annotations.

We implemented the universal virtual image printer driver under Microsoft's Windows operating system, which is installed in the same way as any physical printer driver. To convert a document to the ImageViewer format, one only needs to open the document using the corresponding application, and use the print command of the application to print it to the virtual image printer. Because the virtual printer intercepts at the end of the image rendering process, it is application independent, and can reliably work with any Windows application that supports printing, including the entire Microsoft's Office suite, and many third-party applications such as AutoCAD and Lotus Notes.

IV. Evaluation

We asked 12 undergraduate students in an advanced multimedia course at the Computer Science Department of Stony Brook University to evaluate the usefulness of DataPhone. The DataPhone prototype used for the evaluation is shown in Figure 6. The students were asked to perform two kinds of testing. First, the students were asked to use DataPhone to explain some complicated computer algorithms, such as radix sort, to each other. This test meant to gauge DataPhone's effectiveness in improving sophisticated collaboration over a phone line. The second test was to use DataPhone to call regular phone, and to use regular phone to call DataPhone. The goal of this test was to identify the difference between the DataPhone and regular phone when a DataPhone is used as a regular phone. The following are questions in a survey that the student evaluators need to answer at the end.

1) What is your general impression about the DataPhone setup? Is it useful or not, in what ways?
2) What do you think about the interactiveness and responsiveness of DataPhone's GUI and communication capability?
3) What do you think about the digital pen drawing experience of DataPhone?
4) What features would you like to add/delete?

Fig. 6. DataPhone Prototype

Several conclusions were drawn from results of the survey among this test user group. First, the relatively small screen size of DataPhone, which is the same as PalmPilot's baseline PDA, does not affect its effectiveness. At first we have some concerns about DataPhone's screen size. But from the students' responses, it appears that the screen size of the current prototype is reasonable for the type of data conferencing sessions that can be carried out on DataPhone.

Second, students in general were positively impressed with the new functionalities that DataPhone adds to traditional phones, and showed great enthusiasm in testing the DataPhone prototype. They were particular interested in playing with audio and annotation simultaneously. Some students immediately devised innovated ways to use the DataPhone. For example, they chatted with each other while playing simple games using the DataPhone, and seemed to genuinely enjoy the experiences.

Students appreciated the simplicity and ubiquity of DataPhone. They were also impressed by DataPhone's easy set-up and portability. Most of them believed that DataPhone is useful both in professional interactions, such as lawyers/accounts and their clients, and in day-to-day lives in consumer households.

V. Conclusion

DataPhone represents a well-engineered package with a nice combination of useful features. It supports data conferencing using standard phone lines, and thus requires minimal set-up effort and infrastructural support. Users operate it in the same way as they with regular phones, thus reducing the learning curve to the minimum. The ONLY skill required for using a DataPhone is to dial a number. DataPhone can record data conferencing sessions, both voice and annotations, and thus represents a general form of answer machine and electronic pad. Finally, it includes a universal virtual image printer to exchange documents with PC applications, making it possible to share most PC documents over a DataPhone connection.

Despite a wide variety of functionalities, DataPhone is inexpensive. The hardware cost of the DataPhone prototype is less than $40 USD. With integrated hardware design and mass production, we believe its retail price can be reduced to less than $100 USD, which is comparable to existing cordless phones.

In terms of future work, we are exploring ways to implement similar DataPhone functionalities for cell phones, and to extend the existing DataPhone prototype to support multi-party conferencing calls.

References

[1] ATG. Simultaneous voice and data technology. *http://www.codekit.com/source/projectessentials/documents/whitepapers/simult.pdf*.
[2] BT. Bt videophone. *http://www.bt.com/videophone/index.jsp*.
[3] Cisco. Cisco ip phone 7960g. *http://www.cisco.com/warp/public/cc/pd/tlhw/prodlit/7960_ds.htm*.
[4] Dlink. Be there real time, anytime with i2eye videophone. *http://www.dlink.com/products/voiceservices/dph100/pdfs/dph100_ds.pdf*.
[5] L. W. News. Linux-powered telephone. *http://old.lwn.net/ 1999/features/TouchPhone*, June 1999.
[6] P. H. News. Toshiba launches character phone tegacky. *http://www.phsmou.org/hotnews/tegacky.html*, Feb. 1999.
[7] Polycom. Where voice, video, and data meet - easily! *http://www.polycom.com*.
[8] Rockwell. At commands for rcv56acx, rcv336acx,rcv288acx, and rcv144acx modems reference manual. *http://www.56k.com/inits/rockwell.pdf*, February 1997.
[9] G. Wilson, J. Ostrem, C. Liu, and D. Fulton. Palm os programmer's api reference. *http://www.palmos.com/dev/support/docs/palmos/ReferenceTOC.html*, April 2003.