

A Collaborative Interface for Multimodal Ink and Audio Documents

Amit Regmi and Stephen M. Watt
Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
{aregmi, watt}@csd.uwo.ca

Abstract

With the increased availability of pen-based devices, it becomes interesting to conduct and to archive multi-party communication sessions that involve audio and digital ink on a shared canvas. Collaborative whiteboards do exist today but typically use complex or closed protocols for communication. As a rule, existing whiteboards are not interoperable across multiple platforms and do not support archival of collaborative sessions for later reference or analysis. We explore how various data formats may be used to represent, to transmit, to record and to synchronize ink and audio channels. We find InkML to be a suitable representation to support platform-independent digital ink in a form supporting both transmission and higher-level semantic analysis. To test our ideas we have developed a complete software implementation as a Skype add-on. This has revealed possible improvements to the page and streaming models of InkML.

1 Introduction

Computing and communications technologies are now inseparable. Communication networks rely in essential ways on computers and personal computers are rarely used without being networked. We now use computers for both voice and written communications. At the same time, computing devices are more commonly providing support for digital pen input. We see this with Tablet PCs, PDAs and digital whiteboards.

For certain forms of collaboration, a combination of voice and pen-input is ideal. Collaborative annotation of architectural drawings or working with mathematical formulae are just two examples. We are interested in how to best support this.

We desire a solution with the following characteristics:

- multi-party collaboration, as for a conference call
- support for audio and ink input, with a shared canvas
- no special equipment required

- known data formats for cross-platform support
- use widely available communications infrastructure
- can be used from a range of devices from PDAs to digital whiteboards
- sessions can be recorded and later played back
- recorded sessions are in a form suitable for later processing, document analysis and recognition.

After considering various alternatives, arrived at a design using InkML [1, 2] to represent digital ink, MP3 to represent the audio data and the Skype backbone for communications. While each of these choices has advantages and disadvantages, we elected to use the most widely supported cross-platform options over single-platform proprietary formats and protocols or multi-platform but less-adopted choices. In particular, the wide use of Skype provides immediate access to a wide user base familiar with the operation of the audio software.

The primary contributions of the paper are:

- validation of InkML as a suitable basis for multi-party, real-time collaboration,
- certain specific observations on how InkML should be enhanced before its final form in order to better support real-time, multi-party collaboration,
- the design and public availability of a proof-of-concept, but already useful, package for ink and audio collaboration, satisfying our criteria above and available as a Skype add-on,
- a design for the archival of ink and audio conferences that allows later meaningful analysis of the digital ink.

The remainder of the paper is organized as follows: Section 2 places our work in context, presenting related work and the necessary background on ink formats and the Skype architecture. Section 3 presents our architecture and discusses its more important aspects. Section 4 concludes with a brief discussion and observations and suggests a few directions for future work.

2 Background

2.1 Related Work

Whiteboard sharing is one of the widely used applications in pen computing. In the earliest forms of whiteboards, portability and efficiency of data exchange were not a prime concern. Multimodal systems with pen and voice interfaces were in use in the 1990s and earlier. QuickSet [3] was a multimodal framework used by the US Navy and US Marine Corps to set up training scenarios and to control the virtual environment. Recent work [4, 5] at the HP Labs, describe a heterogeneous collaborative environment for digital ink exchange. There are several other applications that use digital ink and/or audio as input modes. Some of them include applications for classroom presentation and capture [6], meetings [7, 8] *etc.*

A number of Skype add-ons have been developed for collaboration. Yugma (with cross-platform support) [9], Mikogo [10], TalkAndWrite [11], WhiteBoardMeeting [12] are a few that are commercially marketed. These tools support collaboration through screen sharing, audio/video conferencing and digital ink exchange. However, we are unaware of any prior work that archives the ink exchange for later reference and analysis.

Multimodal interactions, including collaboration, are the focus of a W3C working group [13]. A recent thesis [14] provides a summary of issues and current trends in multimodal collaboration by certain vendors.

2.2 InkML

InkML [1] is the Ink-Markup Language that provides a data format for the representation of the digital ink generated by a stylus or an electronic pen. Applications such as handwriting recognizers, paint programs *etc.* can process the digital ink with InkML as the common format. InkML also has a support for a wide range of hardware devices. With the emergence of InkML as an open standard, proposed by the W3C Multimodal Interaction Activity Group [13], a wide variety of applications have been using it for data representation. It is a platform-independent standard that can be used by applications on different operating systems. Hardware and software vendors had previously used proprietary formats for ink representation.

InkML is the medium for digital ink representation in our InkAndAudio multimodal chat application. Although other digital ink formats such as UNIPEN [15], Jot [16] *etc.* do exist, our choice was InkML based on its being an open and an up-to-date standard.

2.3 Skype and the Skype API

We have chosen to use Skype as a backbone for ink and audio collaboration at a distance. Skype is at present the global leader in VoIP and has an increasing user-base with

```
SKYPE4COMLib.Skype objSkype;
SKYPE4COMLib.Call objCallOne, objCallTwo;
objSkype = new SKYPE4COMLib.Skype();
objSkype.Attach(7,true);
objCallOne = objSkype.PlaceCall(
    "test","", "", "");
while(objCallOne.Status != SKYPE4COMLib.
    TCallStatus.clsInProgress) { }
objCallOne.Hold();

objCallTwo = objSkype.PlaceCall(
    "echo123", "", "", "");
while(objCallTwo.Status != SKYPE4COMLib.
    TCallStatus.clsInProgress) { }
objCallTwo.Join(objCallOne.Id);
```

Figure 1. Conf. call using Skype4COM in C#.

over 300 million registered users [17]. Its application interface is familiar to a large population and presents a number of convenient features.

Text-based cross-platform API provided by Skype works based on Operating System messaging with Windows Messaging on Windows; Cocoa, Carbon or AppleScript in Mac OSX and X11 or D-BUS messaging in Linux environment. The Skype public API has been exported in various forms such as Skype4COM, Skype4Py and Skype4Java. Such pre-built libraries make it easier to handle the native Skype API calls written in C++. Skype4COM was our choice for the Windows client implementation as it could be used in any ActiveX environment, such as Visual Studio, Delphi *etc.* Figure 1 illustrates an example C# code that uses Skype4COM.

3 System Architecture

Most of the available whiteboard applications used for collaboration are built on their own complex protocols for data exchange [18, 4]. Instead of re-inventing a new protocol, our system design takes the Skype peer-to-peer protocol as the data exchange back-end after a careful analysis on its advantages. The Skype client can run on different mobile devices and on various platforms. The availability of public API to interact with the Skype network makes it a good choice for the implementation.

3.1 InkAndAudio Client for Windows

Microsoft's Tablet PC SDK had features to support digital ink for Windows. Since its first release in 2001, it has been improved significantly. Windows Presentation Foundation (WPF), on the other hand, is a graphical subsystem in the Microsoft .NET framework 3.0. Digital ink collection and rendering, functions traditionally present only on the Tablet PC platform have been incorporated into WPF as first class member of the framework. The InkCollector, InkPicture and InkOverlay controls found in the classic Tablet PC

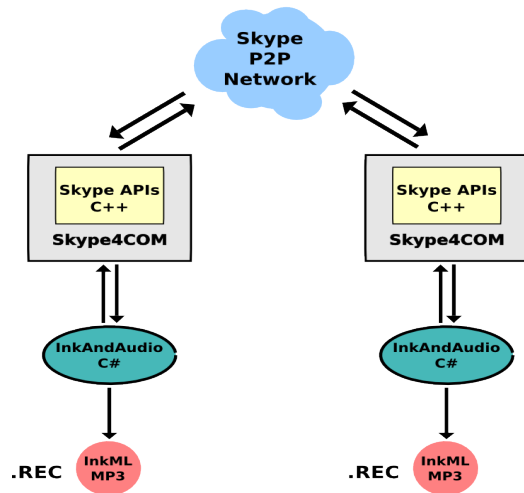


Figure 2. The InkAndAudio Chat application.

platform API are put together in the *<InkCanvas>* element with improved functionalities.

Figure 2 shows how two participants communicate with InkAndAudio Chat using our framework. For our implementation, InkAndAudio Chat handles the data exchange using the App2App communication provided by the Skype4COM wrapper.

An InkCanvas element serves as a whiteboard that is shared among chat participants. Though digital strokes do not occupy as much volume as audi/ data, real-time stroke transmission needs to be ensured. Care should be taken to set a proper size for the data packets and suitable frequency of data transfer. Sending small chunks of data more frequently gives better results than sending large chunks of data in wide intervals. Long traces drawn are to be broken and transmitted in parts while sharing the whiteboard. Such long strokes can be stored as short “*continuation*” strokes in an InkML file and grouped together into a *<traceGroup>* element.

Storage and processing of audio involved in the chat session is handled by the *clsWaveProcessor* class [19]. For our implementation, it has been modified to handle 16000 Hz 16 bit mono audio, the standard Skype audio format. It combines the Skype audio stream with the microphone input and produces a single uncompressed WAV file.

The LAME encoder [20] does further compression on the *clsWaveProcessor*-generated WAV file to produce an MP3; a format widely supported across platforms.

3.2 PyInkAndAudio Client for Unix

Ink application development for Mac OS X and Linux becomes challenging when such operating systems lack the Tablet PC support that Windows have. In such environments, one has to look for alternate ways to achieve the

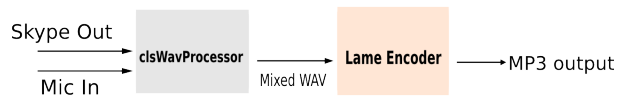


Figure 3. Audio compression with LAME.

```
# python
>>> import objc
>>> objc.loadBundle("SkypeAPI", globals(),
    bundle_path= objc.pathForFramework(
        "/Applications/Skype.app/Contents
        /Frameworks/Skype.framework"))
NSBundle </Users/amit/Library/Frameworks
/Skype.framework> (loaded)
>>> print SkypeAPI
<objective-c class SkypeAPI at 0xd0004140>
>>> SkypeAPI.isSkypeRunning()
1
```

Figure 4. Using PyObjC bridge Mac OS X.

InkCanvas functionality that our Windows client has.

A suitable application development environment for Linux is provided by Tkinter with Python. Tkinter [21] is the most commonly used GUI programming toolkit for Python and is also portable between other environments such as Mac OS X and Windows. Skype applications can be developed quickly with Skype4Py, Tkinter, and Pmw modules.

Cocoa [22] is one of the native application programming environments provided by Apple Inc. for the Mac OS X. Applications developed in Cocoa comply with Apple’s human interface guidelines. Usual Cocoa applications are written using Objective-C language. Now, bridging languages like PyObjC, allow native Mac OS X applications to use the higher level features of Python such as garbage collection and regular expressions. This makes it possible for us to develop Skype applications for Mac OS X, using Cocoa, Skype4Py and PyObjC.

The application development for Linux and Mac OS X face challenges in the way Skype WAV audio is handled. One might have to write WAV mixing algorithms to be able to combine the microphone input and the incoming audio. As our InkAMP player plays .REC files that saves audio as an MP3, there is a need for an audio encoder to convert WAV data into MP3 which is done by LAME encoder in InkAndAudio Chat for Windows. Handling digital ink will be difficult in such environments as one has to simulate the numerous stroke events provided by *<InkCanvas>* in WPF for the Windows client.

PyInkAndAudio is a simple prototype chat client for Linux and Mac OS X that supports only a few of the interaction capabilities available in InkAndAudio Chat client for Windows, but nevertheless was a suitable proof-of-concept. It is based on the Skype4Py [23] multi-platform Skype API wrapper for Python.

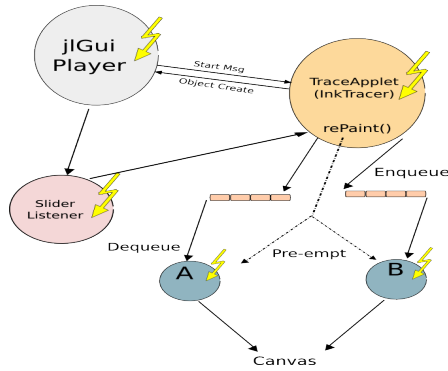


Figure 5. InkAMP's multi-threaded model.

3.3 Chat Session Archival

To store audio and ink together, we use a composite data format. The *.REC* file for a chat session contains the corresponding InkML file and an MP3 audio. The two files are zipped into a single *.REC* archive that may be used for playback by our InkAMP player.

3.4 The Ink and Audio Media Player

InkAMP is an integration of two Java applications, InkTracer and the jIGui music player.

InkTracer. InkTracer is a Java application, developed by colleagues at Ontario Research Center for Computer Algebra (ORCCA). It reads a valid InkML file to capture all the stored stroke information. It then paints the strokes on the application-canvas which is a JPanel component that can be drawn and painted.

The jIGui Music Player. The jIGui [24] is an open source graphical music player written in Java. As a Java application, the jIGui player could be run in a number of operating systems, and could easily be integrated with the InkTracer application. It supports WAV, AU, AIFF, SPEEX, MP3 and other audio formats.

Multi-threaded Model of InkAMP. InkAMP is a multi-threaded Java application whose thread model is depicted by Figure 5. InkAMP can run in any platform that supports Java and operates on the following logic:

- It determines the number of participants N , depending on the number of different GUIDs in the InkML file.
- Creates N painter threads with a common canvas.
- Each *<traceGroup>* in the InkML file is associated to one participant.
- Long traces would already have been broken as “*continuation*” traces by the chat application and stored accordingly.
- One thread reads the InkML and then populates the N queues with the respective traces coming from N participants.

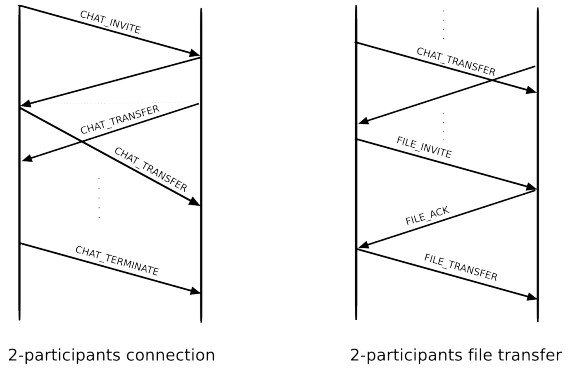


Figure 6. Interaction with 2 participants.

- N painter threads read traces from the respective queues and start painting on the common canvas.

Synchronization. The InkAMP player is a multi-threaded Java application with multiple painter threads, for multiple participants. The synchronization between these threads and the main audio playback thread is based on the global time offset stored by each stroke. The Java threads make use of “Java barrier” pattern for synchronization.

The basic idea was to make the audio thread wait for the painter threads when they lag behind. After each barrier interval (Δt), the audio thread blocks and waits for the painter threads, until they reach the barrier mark. The process repeats until the audio thread completes the audio playback.

3.5 Using the Peer-to-Peer Protocol

The Need for Data Serialization. The primary advantage of serialization of objects in the .NET programming environment is to convert objects into a format that can be transmitted over the network.

Binary serialization gives a better performance because of the smaller volume of the serialized data, whereas, XML serialization gives human readable data with greater flexibility. We have chosen binary serialization over XML serialization, since our aim was to make the stroke transmission faster and efficient.

Establishing Connection. Figure 6 shows how a communication channel is established when 2 participants interact using InkAndAudio Chat. When a user selects an online user from the list, InkAndAudio Chat client initiates the communication with CHAT_INVITE message. It establishes a secure communication channel once it gets an acknowledgment *i.e.* the CHAT_ACK message, from the other party. Once the stream for reading and writing has been set up, `stream.Write()` API is used for stroke transfer with the CHAT_TRANSFER message.

Data Exchange in a Conference. Figure 7 shows the data exchange model for a conference call. The conference is

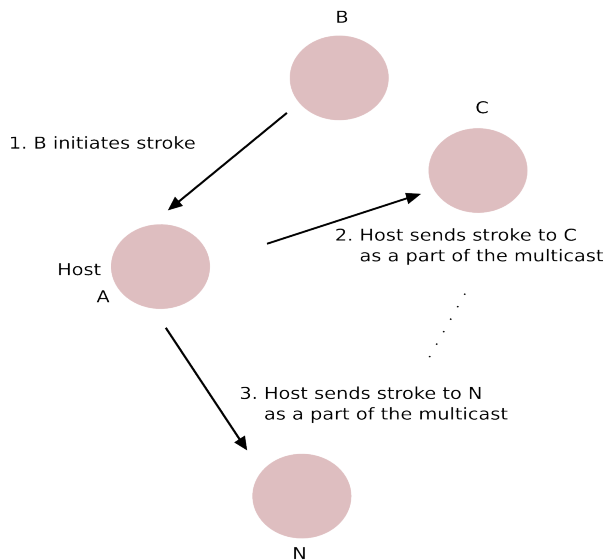


Figure 7. Conference and data multicast.

initiated by the host which has a connection with every other participant. Relay of data takes place through the host node. Audio routing to participating nodes is handled by the Skype client internally, and is written to the corresponding WAV files using the Skype API.

4 Discussion and Conclusions

We have explored the issues in developing a multi-modal ink and audio tool for collaboration. We have focused on cross-platform viability, and open data formats. This allows recorded sessions to be viewed either as voice-annotated documents or as recorded voice sessions with ink annotations. One of our objectives was to test the suitability of InkML to represent streaming ink data in a collaborative setting. We have found that the current description of “Streaming InkML” makes the assumption that strokes can be totally ordered in time. This is in general not true if there are multiple writers. Since InkML is intended to support collaboration on shared canvases, the draft standard should be improved. Secondly, we have found that during InkAndAudio Chat sessions participants will clear the canvas to start a fresh topic. We believe this is close to function of a page model, which InkML lacks.

As a practical matter, we have presented the design and implementation of InkAndAudio Chat. Also, PyInkAndAudio was introduced as a chat client for Linux and Mac OS X as a proof-of-concept to show cross-platform support provided by our multimodal application framework.

There are a number of possible future directions for this work. We would like to understand how to best integrate specialized analysis and recognition components and apply them to parts of sessions. In particular, we wish to handle mathematics and sketch recognition. Our system can also be extended and integrated with external computation servers in a manner similar to what Electronic Chalk-

board [25] does with the Maple computer algebra system and the Mathematica system. Such capabilities would be useful for distance learning applications or even face-to-face collaboration in the classroom.

References

- [1] Y.-M. Chee, M. Froumentin, and S. M. W. (editors). Ink Markup Language (InkML), October 2006. <http://www.w3.org/TR/InkML/>.
- [2] S. Watt.?. Proc. ICDAR 2007, pp 457–460, 2007.
- [3] P. R. Cohen *et al.* QuickSet: multimodal interaction for distributed applications. In *MULTIMEDIA '97: Proc. fifth ACM international conf. on Multimedia*, pp 31–40, 1997. ACM.
- [4] M. P. A. M. Selvaraj and S. Madhvanath. Peer-to-peer ink messaging across heterogeneous devices and platforms. In *Compute '08: Proc. 1st Bangalore annual Compute conf.*, pp 1–5, 2008. ACM.
- [5] HP Labs. Sharing Digital Ink in Heterogeneous Collaborative Environments. <http://www.hpl.hp.com/techreports/2008/HPL-2008-54.html>.
- [6] G. D. Abowd *et al.* Classroom 2000: a system for capturing and accessing multimedia classroom experiences. In *CHI '98 conf. summary on human factors in computing systems*, pp 20–21, 1998. ACM.
- [7] E. R. Pedersen *et al.* Tivoli: an electronic whiteboard for informal workgroup meetings. In *CHI '93: Proc. INTERACT '93 and CHI '93 conf. on human factors in computing systems*, pp 391–398, New York, NY, USA, 1993. ACM.
- [8] M. Stefik *et al.* Beyond the chalkboard: computer support for collaboration and problem solving in meetings. *Commun. ACM*, 30(1):32–47, 1987.
- [9] http://www.yugma.com/share_skype.php
- [10] <http://www.mikogo.com>
- [11] <http://www.talkandwrite.com>
- [12] <http://www.whiteboardmeeting.com/>
- [13] W3C Multimodal Interaction Working Group. <http://www.w3.org/2002/mmi/>.
- [14] M. Zia Hyardi, Unified Communications: Convergence of Platforms and Strategies of Two Software Vendors. M.Sc. Thesis, MIT, 2008.
- [15] I. Guyon. Unipen 1.0 Format Definition. The Unipen Consortium, 1994. <http://www.unipen.org/dataformats.html>.
- [16] Slate Corporation. JOT - A specification for an Ink Storage and Interchange Format. <http://unipen.nici.kun.nl/jot.html>.
- [17] Skype Affiliate Program. <http://Skype.com/business/partners/affiliate/>.
- [18] H. Ning *et al.* InkBoard - Tablet PC Enabled Design oriented Learning. In *CATE*, pp 154–160, 2004.
- [19] CodeProject. Free source code and programming help. http://www.codeproject.com/KB/cs/WAVE_Processor_In_CSharp.aspx.
- [20] The LAME Project. LAME Ain't an Mp3 Encoder. <http://lame.sourceforge.net/>.
- [21] Python.org. Tkinter - Python interface to Tcl/Tk. <http://docs.python.org/library/tkinter.html>.
- [22] Apple Inc. Cocoa: The Objective-C Programming Language. <http://developer.apple.com/>.
- [23] Skype Limited. Skype Developer Zone. <https://developer.skype.com/wiki/Skype4Py>.
- [24] JavaZOOM. jGui : Music Player for the Java Platform, 1999. <http://www.javazoom.net/jlgui/jlgui.html>.
- [25] G. Friedland *et al.* Teaching with an intelligent electronic chalkboard. In *ETP '04: Proc. 2004 ACM SIGMM workshop on Effective telepresence*, pp 16–23, 2004. ACM.