A Bidirectional Haptic Communication Framework and an Authoring Tool for an Instant Messenger

Youngjae Kim[1], Heesook Shin[2], and Minsoo Hahn[1]
[1]Digital Media Lab, Information and Communications University, 517-10 Dogok-dong, Gangnam-gu, Seoul, 135-854, Korea
[2]Wearable Computing Research Team, ETRI, 138 Gajeongno, Yuseong-gu, Daejeon, 305-700, Korea
youngjae@icu.ac.kr, hsshin8@etri.re.kr, mshahn@icu.ac.kr

***Abstract* — Haptic is a promising interface modality for the non-verbal expression. Since an instant messenger is an intimate conversation method for computer users in these days, there are several research trends [1] to combine instant messenger with a haptic device. This paper aims to provide a haptic communication framework for achieving three goals: 1) To author a sequence of actions of versatile haptic devices. 2) To receive haptic-emoticon (we call TouchCon) from network peer. 3) To send TouchCon by user command or sensor input. With this system, user can do chatting with not only text and bitmap emoticons, but also user-composed haptic actions.**

***Keywords* — Haptic, Emoticon, Authoring Tool, Instant Messenger, Multi-modal.**

1. Introduction

As the email and instant messenger are widely used as one of a general emotional expression today, these emoticons are going to spread into a various format. The evolution of emoticon follows by an evolution of information technology. In other words, text-based emoticon is created by console-based email environment, graphical emoticon comes after the GUI is widely adopted, and animated multimedia emoticon is getting popular as the internet environment is sufficient to display small video clips without any trouble. At this moment, we can question to ourselves what the future emoticon behaves like. We can imagine the form of the future of emoticons as follows; it will adapt to various hardware form factor, ubiquitous network with mobility, and multi-sensor multi-actuator expression.

Meanwhile, the term 'haptic' was not familiar with people until year 2005. As a touch screen mobile device is getting popular, some of them adopt VibeTonz [7] technology which enables vibration motor while the touch screen is tapped. Some mobile phone vendors start to use the term 'haptic' in commercial so that many people can understand the effects of haptic and their related experience.

In this paper, we propose the haptic controller system which brings haptic experience to user of instant messenger. Our contribution can be summarized into following three points: First, we design a framework to edit a behavior of both sensors and actuators for bidirectional haptic communication. Second, we implement timeline-based authoring tool for everyone can compose his or her own haptic expression easily. Finally yet importantly, we define three XML schemas for dealing with various haptic devices and TouchCon interaction.

2. Related Work

Today's IM users tend to create and share individually-created emoticon. Previous work [2, 3] developed authoring tools similar to what we propose here, however, it cannot deal with input (sensor) data and editing is difficult for computer novices. In addition, research on multimodal framework [4] is limited to graphical expression, so it cannot send customizable haptic information to the peer.

HAMLET [5] is an authoring tool to compose HAML. HAML is a haptic markup language. This work supports MPEG standard XML schema and can be mixed with graphic image and haptic information. However, it is specialized to 3D graphic manipulation with PHANTOM control device and it does not aim to timeline-based haptic interaction composing or instant messenger environment. So far, very little has been done in both bidirectional haptic communication or in its authoring tool.

3. Design of Haptic Communication Framework

Toward a bidirectional haptic communication, our proposed system needs following requirements.

First, our system has to support various form of hardware types which consists of sensor and actuator. We assume that the future of instant messaging is done by many types of device. For enriching haptic experience, our system needs to support the execution of the haptic emoticon by clicking the graphcal icon or stimulating the sensors.

Second, our system requires compatibility with previous instant messenger application. Since our framework runs as a plugin type of an instant messenger, it should have software interface (a communication handler).

Third, each haptic emoticon can be created and be modified easily by user. By providing well-designed WISIWIG style authoring tool, each haptic emoticon should be managed simply. Our system supports both framework and timeline-based authoring tool.

The figure 1 is the overall architecture of our framework. From left to right, a library object exposes interface to communicate with user programs and hardware object manages sensors and actuators.

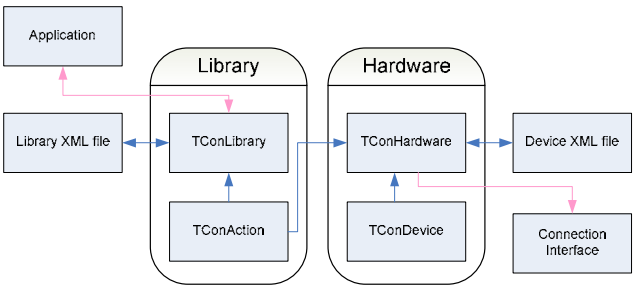

Figure 1. Overall architecture of TouchCon framework

The prefix 'TCon' is a short name of TouchCon. TConLibrary is a set of TConAction and TConHardware is a set of TConDevice.

The following table describes each component and its specification.

Table 1. Description of each component

Component Name	Description
TConLibrary	A list of TouchCons
TConAction	A TouchCon which composed by a user or a haptic emoticon distributor
TConHardware	A list of a connected TouchCon hardware
TConDevice	An hardware-side protocol description of a single
Application	TouchCon deivce (generally an actuator or a sensor)
Library XML file	An XML file which stores composed TouchCons
Device XML file	An XML file which stores hardware protocol specification and its capable commands
Connection Interface	A method to communicate with a TouchCon hardware. It can be a legacy serial port or a USB interface.

As we can see in Table 1, the TouchCon framework requires two XML schema; *Library XML* and *Device XML*. The proposed system is designed to provide a departed domain to both haptic designer and hardware developer. Once a hardware developer creates a haptic device, a developer lists all available commands of the device. For example, if the device has one vibration motor, the command may have five level of vibration power (we say five properties in one device). Or, a developer may allow only one vibration activity. These device-specific commands are the sources of what haptic designer creates a TouchCon. When a designer creates a TouchCon named 'Smile,' then he or she uses available device commands and compose with time and properties.

We invite two XML schemas into our framework for flexibility, readability, and compatibility; Flexibility is the first advantage of our system. In Ubiquitous computing environment, the haptic communication might be expressed to various hardware such as a mobile, fridge, or a car seat. With Device XML file, regardless of any complex hardware, the TouchCon framework can control as developer describes with no trouble.

The advantage of readability helps both developer and haptic designer. Under our inspection, both hardware developer and haptic emoticon designer easily edit xml files using simple Notepad application. This proves that our XML schema brings opportunity to manipulate without any special application like TouchCon authoring tool.

Finally, in regard of compatibility, we assume that each user may have different haptic device. Under this consideration, although a peer's device is not equivalent to user's device, our system has to make TouchCon interaction available. In other words, TouchCon itself is not limited or is not designed only for one device. If a composed TouchCon has same name and same action command, not a device command, the TouchCon can run even if the peer has a different form of haptic device. For example, a user does IM by a laptop and the peer does by a mobile phone. While the laptop expresses 'Smile' TouchCon by blinking LEDs on the screen, the mobile phone expresses it by vibration. In addition, the compatibility has another meaning. The TouchCon-related XML file can convert to HAML and other specially designed haptic interaction specification. By using understandable XML schema, a developer can make conversion software easily.

```
<?xml version="1.0" encoding="utf-8"?>
<TCons User="default">
    <TCon Name="Smile" Image="01.bmp" Speed="1" Description="">
        <Action Device="UpperLip" StartTime="0" Duration="30" Property="Purple" />
        <Action Device="LowerLip" StartTime="50" Duration="30" Property="Purple" />
        <Action Device="Vibrator" StartTime="100" Duration="50" Property="7_111111" />
    </TCon>
  <TCon Name="Kiss" Image="u_121.bmp" Speed="1" Description="">
    <Action Device="UpperLip" StartTime="0" Duration="50" Property="Red" />
    <Action Device="LowerLip" StartTime="50" Duration="30" Property="Purple" />
    <Action Device="Vibrator" StartTime="0" Duration="100" Property="3_111111" />
  </TCon>
</TCons>
```

Figure 2. An example of a TouchCon Library XML file

The Library XML file is a set of each TouchCon action. As we can see in figure 2, one TouchCon action means one emoticon of TouchCon. It has 4 attributes and extensible Action part. The name of TouchCon is mandatory and others are optional. Going down a deeper level, one Action part has 4 attributes. Each of which tells a device to use, starting moment, duration to activate, and property value to command.

Figure 3 shows an example of TouchCon Device XML file. It describes three devices.

```
<?xml version="1.0" encoding="utf-8"?>
<TConDevices>
    <TConDevice Name="UpperLip" Output="True" Description="Mouth Upper Lip" DataType="String">
        <Property Name="Red" Start="U1AY" End="U1AN" />
        <Property Name="Green" Start="U2AY" End="U2AN" />
        <Property Name="Yellow" Start="U3AY" End="U3AN" />
        <Property Name="Blue" Start="U4AY" End="U4AN" />
        <Property Name="Purple" Start="U5AY" End="U5AN" />
        <Property Name="Cyan" Start="U6AY" End="U6AN" />
        <Property Name="White" Start="U7AY" End="U7AN" />
        <Property Name="Off" Start="U0AY" End="U0AN" />
    </TConDevice>
    <TConDevice Name="Pin" Output="True" Description="Pin" DataType="String">
        <Property Name="Left" Start="P1AY" End="P1AN" />
        <Property Name="Right" Start="P2AY" End="P2AN" />
        <Property Name="Both" Start="P3AY" End="P3AN" />
        <Property Name="Off" Start="P0AY" End="P0AN" />
    </TConDevice>
    <TConDevice Name="Heat" Output="True" Description="Heater" DataType="String">
        <Property Name="On" Start="H1AY" End="H1AN" />
        <Property Name="Off" Start="H0AY" End="H0AN" />
    </TConDevice>
</TConDevices>
```

Figure 3. An example of a TouchCon Device XML file

The Device XML file is created by a hardware developer. TConDevice has four attributes. The output attribute stands for a direction of interaction. When the value of output is true, it means the device is an actuator. If not, it means sensor for retrieving data from sensor device to the TouchCon

framework. With this feature, our system is able to invite multi sensor and multi actuator device.

Since our work can send TouchCon triggered by sensor inputs, this is why we call the system bidirectional. When a sensor captures an event of input and the sensor value is matched with user-defined sensor value, our framework prompts one of the followings: activate user-side TouchCon, transmit a TouchCon to the peer, or do nothing. Moreover, the received TouchCon from a peer is automatically added to a library, so user can disperse thereafter easily.

In summary, the TouchCon framework has both sensor input and actuator runnings with a single device description. Each TouchCon has

4. Design and Implementation of the TouchCon Authoring Tool

The development of the authoring tool is motivated by typical multimedia editing application such as Adobe Premier and Flash. As these timeline-based tools are now prevalent, we attempt to provide similar look-and-feels and to provide them experience of haptic authoring. For that, we design the tool with its timeline arranged horizontally and haptic device layers vertically. Each haptic device layer denotes every haptic actuator's behaviour running in parallel. Figure 4 below is a screenshot of our TouchCon authoring tool.

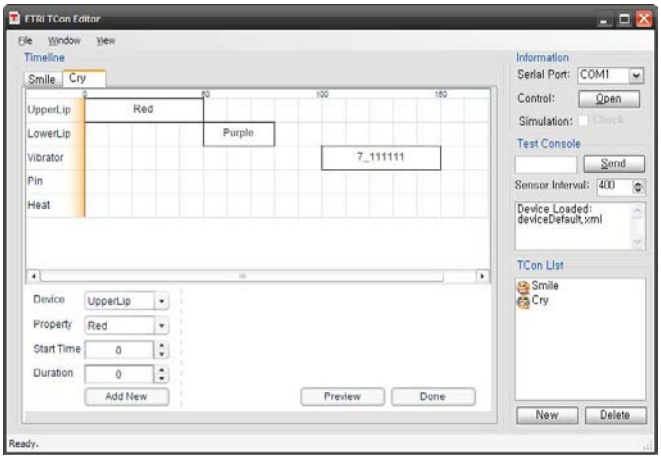

Figure 4. A screenshot of the TouchCon authoring tool

As we can see in Figure 4, a layout and position of components resemble commercialized multimedia editing application for intimate user experience. Before user starts to compose, the TouchCon editor has to load hardware description file. The hardware description file has a set of commands to control connected hardware.

Using the authoring tool is simple; select device and its property, arrange time, and save with appropriate TouchCon name. To describe detail, a designer, a user of this authoring tool, sees each device list in the left side; currently five devices are listed including UpperLip or Heat. Once the designer selects one device, property is refreshed. These two values are given by the device developer. The start time and duration can be defined in millisecond. When the developer clicks Add New buttons, new action is added in the timeline. A preview button provides user to test current TouchCon. The result is saved into Library XML format.

The right side of the authoring tool shows the output of the device and connection settings. In the bottom right side of the software named TCon List, it lists up all TouchCons.

5. Evaluation

We adopt our system with simple instant messenger application and a pair of specially designed hardware device. As we can see in Figure 5, a TouchCon list is floated right side of the messenger window.

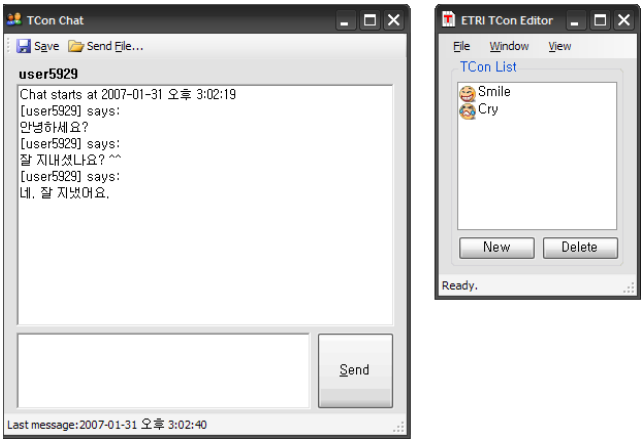

Figure 5. An example of an instant messenger and TouchCon window

User can do double-click the TouchCon list, and the haptic signal is transferred to the peer. If the peer sends new TouchCon through the messenger, it adds new one in the list automatically. With this feature, user-generated TouchCons are to be shared to another user easily. The TouchCon list window can be changed into TouchCon authoring tool directly by simply clicking 'View' menu button. When user clicks the 'View' button, the small window changes into Figure 4, a TouchCon edit mode. It means, the user can create or edit TouchCon while using the messenger. The figure 6 below shows the prototype of a haptic hardware. Two pieces are fabricated for peer to peer instant messenger environment.

Figure 6. A prototype of a haptic hardware

The hardware consists of two palm-rest type part and one lip-shaped part. Figure 7 describes the combination of actuators and sensors components in each module. For test purpose, we designed that the prototype hardware has sensors and actuators as many as possible. The outfit of hardware is molded with silicon for soft touch feeling. With this silicon material and design, user can have comfortable and intimate feelings. Figure 8 shows how to use the prototype hardware. As we can see, the use put his or her hand on the palm-rest part in order to feel the actuator.

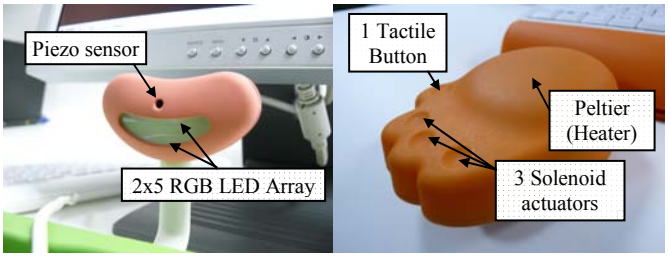

Figure 7. Actuators and Sensors are inserted into each hardware part

Figure 8. A usage of prototype hardware

User can send a TouchCon by touching lip-shaped part (see right picture of Figure 8). As we described earlier in Chapter 3, user can define how the touch sensor affects. It can be defined into three type; send one TouchCon, Activate user's hardware, for example blinking the LEDs, or do nothing. Figure 8 shows the touch behavior turns user's LED's on the lip-shaped part.

The research of effectiveness of TouchCon communication with above decive can be found in the earlier article [6].

6. Conclusion

We design and implement the framework which enables various haptic devices to interact through network. In addition, the comprehensive authoring tool helps user to edit their own haptic expression easily. The framework is capable to link with widely used instant messenger such as MSN or NateOn messenger. We hope these compatibility and easy-to-use interface bring enjoyment of haptic communication to the user. We will expand our work to mobile environment and VR telepresence field for allowing people the experience of haptic communication.

REFERENCES

[1] K. Suzuki, S. Hashimoto, "Feellight: a communication device for distant nonverbal exchange.", ACM SIGMM Workshop on Effective Telepresence, 2004.

[2] A.F. Rovers, H.A. van Essen, "Design and Evaluation of Hapticons for enriched Instant Messaging", Eurohaptics 2004, 2004.

[3] M. Rossi, K. Tuer, D. Wang, "A new design paradigm for the rapid development of haptic and telehaptic applications", CCA 2005. IEEE Conference on, Vol., Iss., 28-31 Aug. 2005, pp. 1246- 1250.

[4] C.L. Lisetti, F.Nasoz, "MAUI: a multimodal affective user interface.", MULTIMEDIA '02. ACM Press, 2002, pp.161-170.

[5] M Eid, S Andrews, A Alamri, A El Saddik, "HAMLAT: A HAML-Based Authoring Tool for Haptic Application Development," LNCS, Vol. 5024/2008, Springer, 2008, pp.857-866

[6] H. Shin, J. Lee, J. Park, Y. Kim, H. Oh, T. Lee, "A Tactile Emotional Interface for Instant Messenger Chat," LNCS, Vol. 4558/2007, Springer, 2007, pp.166-175

[7] Immersion Corporation, VibeTonz System Product, http://immersion.com/mobility/docs/VibeTonz_System_oct07_v3lr.pdf, 2007