

Bridging Digital and Physical Worlds Using Tangible Drag-and-Drop Interfaces

Mathieu Hopmann, Mario Gutierrez, Daniel Thalmann, and Frederic Vexo

VRLab, EPFL, 1015 Lausanne, Switzerland
{mathieu.hopmann,daniel.thalmann}@epfl.ch,
{mgutierrez,frederic.vexo}@logitech.com
<http://vrlab.epfl.ch>

Abstract. The last ten years have seen an explosion in the diversity of digital-life devices, e.g. music and video players. However, the interaction paradigm to use these devices has remained mostly unchanged. Remote controls are still the most common way to manage a digital-life device. Moreover, the interaction between devices themselves is still very limited and rarely addressed by a remote control interface. We present in this paper a study of *tangible drag-and-drop*, a remote control interface based on the well-known paradigm coming from the graphical user interface. This interaction technique aims at reducing the gap between the digital and physical worlds, enabling the transfer of digital data from one device to another. To validate such a concept, we present two prototypes, along with user studies and a general discussion about the tangible drag-and-drop technique.

Keywords: Interaction techniques, Input devices, Tangible interaction, Remote control, Drag-and-drop, Multimedia content.

1 Introduction

At the end of the 90's, digital content at home was limited and mainly stored in home computers. But with the explosion of digital formats and the Internet, our “digital life” has hugely gained in importance. Pictures, songs, movies, almost all of our multimedia content could be dematerialized. The diversity of devices to manage this digital life has also drastically increased: laptops, digital cameras, MP3 players, digital photo frames, Smartphones . . .

A problem resulting from this device multiplication is the content dispersal: having many devices often results on having different content on each device, and synchronizing everything is a baffling problem. For this reason and with the multiplication of devices connected to the Internet, the Web becomes more and more the aggregator of our digital data. We store our emails on a webmail provider, share our pictures on an online photo gallery or on a social network, and listen to music via webradios or music recommender systems. However, the main device to access and manage our digital data locally and on the web remains the same: a computer, a mouse and a keyboard.

Mouse devices, which have accompanied computers for more than 20 years, are still the main device to select, move and interact with digital content. The reason is probably because it combines efficiency, easiness, and low physical effort. Standard interaction tasks offered by the mouse are selection, pointing, drag-and-drop, copy/cut and paste. . . These tasks are now massively adopted by computer users, and we use them everyday in a natural way. A question we asked ourselves was: among these interaction tasks, which are the ones more closely related to everyday life gestures.

Pointing is one of them. Every child points his/her finger to pick up something, and everyday we are pointing devices with remote controls to interact with them. It is natural, but also limited: pointing allows to select, but generally not to perform an action. On the other hand, copy and paste allows to transfer text, files or objects from a source to a destination, but is hardly comparable to an everyday life gesture: we are still not able to clone things right away ! On the contrary, drag-and-drop is an action performed continually in everyday life: when we take an object to put it in a different location, it is kind of a drag-and-drop action.

In human-computer interaction, drag-and-drop is the action of clicking on a virtual object, and dragging it to a different location. This is a task broadly used in graphical user interfaces: it is natural, efficient, and with a low learning curve. That is why we chose to adapt this in a tangible way, and use it to bridge the gap between digital data and physical devices. In this paper, we present an interaction technique, called tangible drag-and-drop, and two prototypes that have been developed in order to evaluate our concept. The first one is intended to ease the interaction when sharing music with friends, whereas the second one aims at managing a multi-room audio system. We describe for both of them the implementation and the feedback we got from users. The paper ends with a discuss about the tangible drag-and-drop experience.

2 Related Work

Despite the fact that our digital life has hugely gained in importance over the last years, the mechanisms to interact with it have poorly evolved. It is still difficult to share digital content between devices and to interact with it. In the smart home[2], which could be a vision of our future home, our environment is surrounded of interconnected devices, which respond smartly to our different actions. This interconnection gives us the possibility to create new kinds of interaction between the different devices. The goal of these new interaction concepts could be to make our digital life more tangible, but also to add efficiency.

The concept of using a tangible user interface to facilitate the manipulation of digital content has been introduced in the middle of the 90's by Fitzmaurice et al. with Bricks[4]. With Bricks, they presented a "Graspable User Interface" to

control virtual objects through physical handles. Ishii and al. continued in this way with Tangible Bits[9], where they suggested to bridge the gap between bits and atoms using graspable objects.

Several works put forward the idea of taking advantage of a tangible user interface in order to manage digital multimedia content. About music, Alonso et al. created the MusicCube[1]. Users interact with the MusicCube using gestures to shuffle music and a rotary dial for song navigation and volume control. With iCandy[5], Graham et al. designed a tangible interface to restore the benefits of physical albums for digital music. About pictures, Nunes et al.[13] implemented a system that allow people to link digital photo with physical souvenirs. In [8], Hsu et al. designed a tangible device which supported gestures to browse digital pictures.

Another aspect of our multimedia life which suffers of the lack of tangibility is the communication between the devices. Many remote controls have been commercialized in order to interact with several devices¹², and the idea of the universal remote control has more than 20 years[15]. However, the goal of these current universal remote controllers remains basic: to replace several remotes by a single one. There is no possibility of interaction between the different devices, the goal is only to limit the number of remote control or the number of steps you need to perform an action (using an "Activity" based remote).

The concept of enhancing our digital life in the "smart home" has been addressed in [17][16] with CRISTAL, a system to simplify "the control of our digital devices in and around the living room". In these papers, they described a system based on an interactive multi-touch surface and a camera. Users can interact with the different devices in the living room by manipulating their representation on the screen. It is a interesting solution for mixing digital files and physical devices on a single surface, but cumbersome and invasive, having a live camera in the living room may be unacceptable for some users.

Different works suggest to use tangibility to facilitate the transfer of digital data between devices. In [14], Rekimoto et al. pointed out two main problems of using multiple computers: the input device, which is often tethered to a single computer, and user interfaces, which are not designed to manage a multiple-computer environment. With pick-and-drop, they tackled this problem by using a pen to pick a virtual object on a screen and drop it on another display. A close idea is described in [18] with Slurp, an eyedropper with haptic and visual feedback which extracted digital media from physical objects in order to inject it into other devices. In [11][10], the authors suggested to use hand gestures in order to connect devices and transfer files from a device to another. In [7], we presented a first tangible drag-and-drop prototype focused on digital pictures and the interaction with a digital photo frame. This first prototype gave encouraging results in terms of interaction between multiple devices, and for this reason we decided to further further develop this concept. This paper presents our latest results.

¹ http://www.logitech.com/index.cfm/remotes/universal_remotes

² <http://www.remotecontrol.philips.com/>

3 Tangible Drag-and-Drop

Drag-and-drop in computer graphical user interfaces (GUI) is intended to drag virtual objects from a location to another in a natural way. However, drag-and-drop is not limited to the action of moving files to a new location. In GUI, you can drag a window border to increase its size and drop it to validate your choice. You can drag a file on a program icon to execute an action on this file. You can drag a picture from a web browser to the desktop in order to create a file which contains this picture. Besides being a fast interaction task to adopt, drag-and-drop is flexible and allows for diverse functionality.

Our idea is to take advantage of this efficient and easy-to-learn interaction task and adapt it to the physical world. Using a ‘drag-and-drop remote control’, we want to be able to select digital content, drag it physically with the remote and drop it on another physical device.

In order to illustrate our idea, we designed two prototypes. The first one is dedicated to share music with friends, the second is intended to manage a multi-room audio system.

3.1 Prototype #1: Sharing Music by Drag-and-Dropping between Devices and a Virtual Shelf

Despite the fact that digital media has already invaded our homes, interaction models have not changed: it is still hard to interact with our music collection without a computer. Our main goal is to make this interaction more simple and efficient, even for people with few computer skills. We focus this first prototype on three points:

- Browsing our music collection in a familiar way
- Sharing music between friends
- Sharing music between devices

In order to browse our music collection, we choose to display it in a familiar context: as we had CDs at home stored in a shelf, our application displays albums on a screen inside a virtual shelf. The user doesn’t need to enter a login or a password. Indeed, the application detects the user’s cell phone, scans the music database and displays her collection inside the virtual shelf on the TV screen. With the remote control, she can choose the album he wants to play, and drag-and-drop it on the sound player close to her and listen to it. If a friend enjoys the music, she can use the application too: a second virtual shelf appears on the TV with her music collection, she just has to select the album in the other shelf and drop it in her collection.

Our application adapts the drag-and-drop interaction task to transfer music files from the digital to the physical world (from the user’s music collection to the selected sound player) and within digital worlds, allowing music sharing between friends.

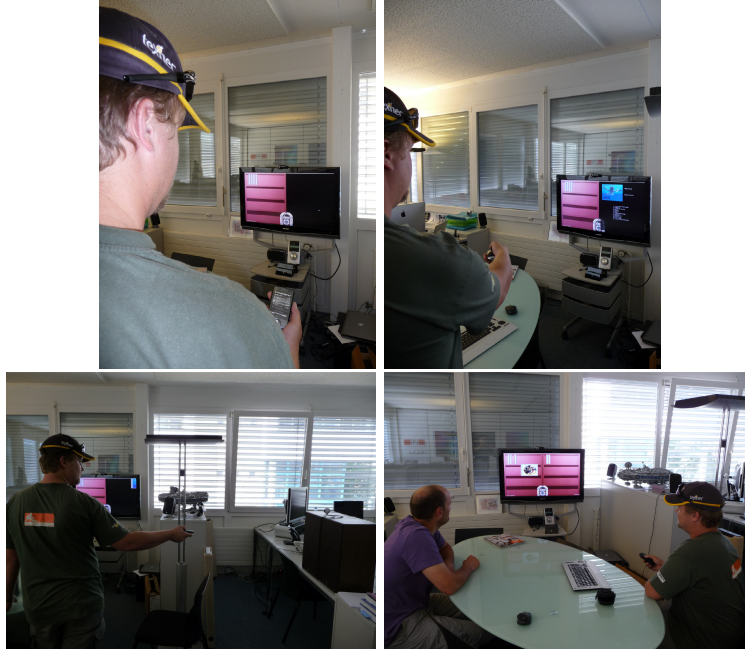


Fig. 1. From top left to bottom right: (I) User's cell phone is detected, a shelf appears with his music collection. (II) The user is browsing his music collection using the MX Air mouse. (III) The user drops an album on the sound player after selecting it in the virtual shelf. (IV) Two users are sharing albums from their music collection.)

Music centralization. The storage of digital music is a real challenge. Keeping our music collection ordered in a single location requires too much time. With this in mind, we decided to use a single place to store user's music collection. For our prototype, we have stored all the music files on a web server, to make it accessible from everywhere.

We have created two MySQL databases to manage the music collections: the first one stores all the basic information about songs (band, album, cover, song name, song number) and their location (a hyperlink to the song path on the server); the second one stores information about users (first name, last name, bluetooth MAC address of the cellphone) and the songs they have in their music collection (a reference to the first database). This is a technical choice above all: sharing music with a friend just consists in adding an entry in the database. In the discussion section, we get back to that point based on feedback from the testers.

Hardware description. The user interacts with our application using a remote control, which is a custom Logitech MX AirTM. This mouse is a cordless air mouse which maps hand motions to cursor control: the mouse follows the user's hand, using a gyroscopic sensor and an accelerometer. This functionality is interesting to control the mouse pointer on a TV for example, from the sofa.

We have stuck an infrared LED under the mouse in order to detect where the user is pointing the mouse. A webcam connected to a computer near the sound player is necessary to detect the mouse. Technically, the sound player is connected to this additional computer, but this is invisible to the user. The user must have the feeling of pointing the shelf and dropping the digital CD in the sound player directly.

At last, the virtual shelf computer has bluetooth capability in order to detect user's bluetooth cell phone.

User identification. To access to her music collection, the user doesn't need to enter a login or password: she is identified with her bluetooth cell phone. As almost everybody has a cell phone, it seems to be the best way to identify transparently the user. To allow this possibility, a Java software is running on the virtual shelf computer. It permanently calls a linux application (hcitools) in order to scan the bluetooth devices in the room, and if a new one is detected, the Java application sends the bluetooth MAC address to the flash-based virtual shelf application using a TCP connection. The virtual shelf application checks if the user is present or not inside the centralized music database. If yes, a new shelf appears with the user's music collection. We limited the application to two shelves (thus, two users): firstly for a size reason, and secondly because it seems useless to allow more users on a single screen (sharing music is often between two persons).

Browsing music. We have designed the virtual shelf to be simple and familiar, with albums stored inside the shelf. Browsing a musical collection is really easy, even for people with limited computer skills. At this point, we have mainly worked on facilitating the browsing and we have taken advantage of the MX Air: pointing a CD with the mouse cursor displays information about it (album and artist name, songs in this album, CD cover), whereas clicking on the CD animates it as if you remove a CD from a real shelf. Then you can share your CD with your friends, listen to it, or release it and an animation replaces the CD at its initial position.

As we explained before, the music information is stored on a distant database, and not directly on the computer. And because the virtual shelf has been developed in flash, it could be run on other computers with a web browser and flash capabilities (the only additional part to install is the bluetooth java module to detect the user's cell phone). Furthermore, new interactive TVs such as GoogleTV include a Web browser with the Adobe Flash plug-in, this enables virtual shelf to directly run on a TV without any additional computer.

Sharing music. Sharing our music with our friends is not an easy task nowadays. You need to parse the music files as any other file type, and sometimes it is even impossible (with ipod or iphone for example if you are not on the computer synchronized with the device). With our application, you can drag-and-drop music from your friend's music collection to your own music collection: you pick up a CD from her shelf and release it on your shelf, and the copy is done. Visually, it appears as a cloning: when you release the mouse button, a new CD

appears under the mouse cursor, and goes next to your other CDs, whereas the original CD goes to its initial position. Technically, the virtual shelf application adds new entries in the user's music database, which are the same entries already present inside the other user's music database. This way, the copy is done instantaneously: the music files are not duplicated, only the database entries are.

Playing music. To play the CD selected on the shelf, the user has two possibilities:

- Dropping the CD on the jukebox displayed between the shelves.
- Spatially drag-and-dropping the CD from the flash application to the physical sound player

For the first point, the virtual shelf application just plays the CD on the TV speakers. There is no technical challenge with this solution: Flash has sound capabilities and could play distant MP3s without any problem. So when a user drops a CD in the jukebox, we play the album by sending the MP3 hyperlink of the first track to the Flash application. The mouse wheel allows to change the album tracks, and the mouse click on the jukebox will turn off the current music.

The second point was more challenging to implement. In a previous work [7], we already developed an application to transfer files from TV to a digital photo frame. We have adapted our software to the virtual shelf application. An additional application dedicated to the webcam image analysis is running on the sound player computer, and informs the virtual shelf application about it. If the user selects a CD and releases the button when the mouse is pointing to the sound player device, then the sound player computer plays the corresponding album. When the sound player is on, the user can stop the music by pointing the device with the mouse, clicking on the left button and releasing the button outside of the webcam area: this way, the user drag-and-drops the sound outside of the music player. Technically, in order to perform the mouse LED detection, we have used `cvBlobsLib`, a blob detection API based on the OpenCV API. Coupled with an infrared LED and a webcam that filters visible light, detecting if the user is pointing or not the mouse in front of the sound player is functional.

3.2 Prototype #2: Tangible Drag-and-Drop to Manage a Multi-Room Audio System

The dematerialization of the music content brings advantages: you can carry all your music collection on a single device, which could be your mp3 player or your cell phone for example. Digital music allows you to have access to your entire music collection everywhere. Paradoxically, playing music at home has not changed that much. People who want to play their digital music are often using computers to do that, and so they are limited to one room.

Different commercial audio systems are now widely available to enjoy a digital music collection in different rooms. For example, Philips with the Streamium

TMwireless music center³ and Logitech with the SqueezeboxTM audio system⁴. Technically, these solutions are fast and easy to deploy, they can play a lot of different digital formats. On the other hand, navigating in these devices is not so comfortable. You have to navigate through a classical linear menu to select artists, albums or playlists, using the buttons on the device or a basic remote control.

Our idea is to use the tangible drag-and-drop technique in order to control a multi-room audio system. Our system is based on the Logitech SqueezeboxTM server for the multi-room audio system, and on the Nintendo WiiTM Remote (also nicknamed Wiimote) for the drag-and-drop remote control. We want to fully control the audio player next to us, but also the players situated in all the other rooms of the house, while sitting in our living room. Thus, we need to be able to:

Browse and select content. The music album collection has to be accessible and browsable. We need to be able to select an album, a web radio or a playlist with our remote control, and to create playlists with it.

Play selected content. After selecting a music content, we could choose to play it on the device in the same room, but we have also the possibility to play it on a further device.

Control the devices. Our remote control has to support all the basic features of a sound system, such as play, pause, next or previous song, etc.

Transfer content. The devices have to communicate with each other. For example, we want to be able to transfer the playlist currently played on a device to another device.

A possible use case could be the following: You are in your living room, sitting in your sofa, and wish to listen to some music. Your entire music collection is displayed on your TV. You select the album you want to play, and drag-and-dropping it from the television to the music player. After some tracks, you want to go to the bathroom to take a shower. Thus, you target your music player, and drag-and-dropp its content to the representation of the bathroom displayed on the screen. The album (or the playlist if you choose several albums) are now playing in the bathroom.

Implementation. This second tangible drag-and-drop prototype is not based on the first prototype. For this experiment, we have selected products which are already available on the market, and we have designed the software to make them communicate with each other.

This approach has the following advantages: Firstly, we did not have to develop a complete multi-room audio system. As explained before, commercial solutions already exist to perform that and they work well. Secondly, users are already familiar with these devices, they wont be facing an unknown environment.

³ www.streamium.com

⁴ <http://www.logitechsqueezebox.com/>

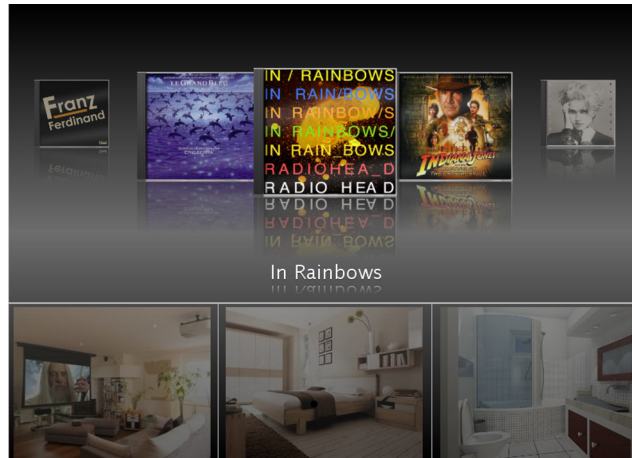


Fig. 2. Graphical user interface of the multi-room audio system



Fig. 3. Concept and hardware of the second prototype

On the other hand, we need the devices to be fully customizable. We choose the Squeezebox™ of Logitech to be the multi-room audio system, and the Nintendo's Wiimote™⁵ to be our tangible drag-and-drop remote control. Both devices can be controlled and customized with open source software libraries.

⁵ <http://www.nintendo.com/wii/what/controllers>

Audio system. Our audio installation is composed of a computer running the Squeezebox™Server, and three Squeezebox™ Radios.

The server provides the access to the music library, and can control the entire system. From the server, we can create a playlist, choose an album/radio/playlist and on which player we want to listen to it, or control the players (play, pause, stop, increase volume, etc.). This is exactly what we want. To communicate with the server, the Squeezebox Server provides a command-line interface via TCP/IP. Commands and queries could be sent using TCP/IP in order to fully control the server. This way, we are able to ask programmatically the album list, the covers, and interact with all the players. We choose to use javaSlimServer⁶, a Java API used to control the Squeezebox Server via the command-line interface, to easily communicate with the server.

Remote control. As tangible drag-and-drop remote control, we choose this time the Nintendo Wii™Remote. The reasons are the following: Firstly, the Wiimote integrates a 1024 x 768 100Hz infrared camera. This is a good solution to detect which device the user is currently pointing at. It favorably replaces the multiple webcams of our first prototype. Secondly, the Wiimote has some output possibilities: it can rumble for a few milliseconds or seconds, there is a speaker so it can play sounds, and there is four LEDs. We take advantage of these Wiimote capacities to give a feedback when the pointed device has changed. Now, when the user moves the remote control in order to point another device, the Wiimote rumbles 100 milliseconds to indicate that the new device has been detected. Finally, another advantage of the Wiimote is that several API already exist to access to the Wiimote abilities. As we are using Java to control the Squeezebox Server, we choose a Java library to communicate with the Wiimote: motej⁷.

The main challenge of this second prototype was to detect where the user is pointing the remote control. Our first idea was to put a different number of infrared LED on the devices. The Wiimote camera can detect four different LEDs at the same time, so we wanted to take advantage of this ability. For example, putting one LED on the television, two on the first Squeezebox Radio, etc., and then identifying which device is targeted depending on the number of LEDs detected. But this solution did not work. At a distance of two meters and more, the Wiimote mixes the different LEDs into a single one. In order to avoid this problem, the different LEDs have to be separated by at least 20 cm. (which is actually the size of the Wii™Sensor Bar). It is acceptable for one or two LEDs, but too invasive for three or four.

Our second idea was to put a single infrared LED on each device, and make each LED blink at a different rate. This was the way we implemented the system. In order to make our LED blink, we use Arduino Duemilanove boards⁸. Arduino is a platform intended to easily create electronic prototypes. We just have to plug our infrared LED on the board and program the blink rate. We equipped each

⁶ <http://code.google.com/p/javaslimserver/>

⁷ <http://motej.sourceforge.net/>

⁸ <http://arduino.cc/>

Squeezebox Radio with an Arduino board and a LED blinking at a particular rate.

The television does not have a blinking infrared LED. Indeed, the user needs to know where she is pointing on the graphical user interface, and we do not want to have a blinking pointer on the screen. For this reason, the screen is equipped with a standard Wii Sensor Bar. This way, there is no device identification problem (the sound systems have LEDs with different blink rates while the screen has a standard infrared LED) and our application can precisely detect where the user is pointing the remote on the screen.

Graphical user interface. The graphical user interface is displayed on the television. This is the central part of our application, as it allows to control all the devices.

Concerning the visualization of the media, we choose to display the albums using the cover flow interface. Cover flow is an interface created by Andrew Coulter Enright[3], but is mainly known for its integration in iTunes 7.0. We select this interface because it is adapted to our prototype environment (television + Wiimote), and also because it is near to the physical representation of a music album. The view is more limited in comparison to the virtual shelf (only 5 albums are visible at the same time), but this interface is less demanding in terms of precision. Indeed, you don't need to point precisely at the album, the selected album is always the one in the center. In order to go to the next or previous album, the user just has to target the television and press the left and right button of the Wiimote.

The first version of the main screen of the graphical user interface was limited to the cover flow. The idea was to display the sound devices only when the drag-and-drop button was pressed. So, the user had to point at the television, select the album through the cover flow interface, press the drag-and-drop button on the Wiimote (at this time, the sound devices appear on the visual interface) and release it when she points at the desire device on the screen. But after some early tests, we noticed that this interface had a major drawback: it did not allow to directly interact with the distant devices (for example, to stop a device that you forget to turn off).

The second version tackles this drawback. We add the devices directly on the main screen of the graphical interface, under the cover flow. In order to represent the sound devices on the screen, we choose to display a representation of the room where they are located. Indeed, it is easier to identify the destination of the selection (bathroom, living room, bedroom) rather than three identical pictures of the Squeezebox Radio with different names. So, when the user wants to interact with the device of a distant room, he/she just has to target the room on the T.V. and press the corresponding Wiimote button.

Controlling the tangible drag-and-drop audio system. To give an idea of how to control the system, the following list partially presents a possible action sequence of the application.

Browse the album collection. Point at the screen and press Wiimote’s left or right button

Play an album. Use the drag-and-drop button (Wiimote’s trigger) to transfer the album from the collection to the physical device or its representation on the screen

Change volume. Point at a playing device (physical or on the screen) and press the plus or minus button of the Wiimote

Transfer content from a player to another. Point at the playing device, press the drag-and-drop button and release it on the destination

Stop playing/clear playlist. Point at the playing device, press the drag-and-drop button and release it on the album collection

4 Evaluation and Discussion

In this section we present the tests we performed to assess the usability and user acceptance of the previously described drag-and-drop applications.

4.1 Evaluation of Prototype 1: Music Sharing

Twelve participants tested our application, and gave their feedback about it. Users enjoyed the simplicity of our application, and the efficiency of drag-and-dropping albums from a collection to another and from the collection to the sound device. During a post-test interview, different remarks and concerns were made about some points of our application.

Browsing on the virtual shelf has been described as really easy and natural. Of course, with this first prototype of virtual shelf, the possibilities are limited, and not adapted for browsing big music collections. Albums are now classified by names, and other criteria should be added. For example, one shelf level could be dedicated to one genre. Creating and managing playlists is another option to add, without reducing the ease of use.

Sharing music between two collections has also been described as really easy and natural, but raised some legal concerns. Our research does not address the legal aspect of content sharing.

Furthermore, our centralized system for music collection could be adapted to deal with these problems: for example, instead of copying the album from one shelf to another, the system could ask to the user if he wants to buy this album on a digital music store. Another example : on a music store with a flat monthly fee (such as the Zune Pass on the Zune Marketplace⁹), subscribers can download unlimited music. Thus, copying a track from one shelf to another could also inform this music store about the transfer, and then adapt the remuneration of each artist depending on the number of downloads.

Finally, our testers have really enjoyed the music transfer from the shelf to the music player using the drag-and-drop paradigm. They described it as natural, convenient and efficient. One of the main lesson we have learned from this first

⁹ <http://www.zune.net/en-US/software/zunepass>

prototype study is that it is possible to mix the on-screen and spatial drag-and-drop without disturbing the user. Our participants used it naturally, going from shelf to shelf or from the shelf to the music device without any problem.

4.2 Evaluation of Prototype 2: Multiroom Audio Management System

In order to get a feedback and evaluate our second prototype, fifteen participants (5 women, 10 men) aged between 25 and 45 (mean 32.5) participated in our experiment. Two-thirds are working in the computer science field, and the others frequently use computers at work or at home (all of them were aware of the drag-and-drop concept in graphical user interfaces). Six of the participants do not use frequently the Wiimote, but they all had at least a little experience with it.

After explaining the tangible drag-and-drop concept and how to control the system, the user had some minutes to play with it and be familiar with the prototype. After this time, we asked them to perform a list of actions:

- Select an album in the GUI and play it on the player in this room (i.e. the living room)
- Change the volume of the device
- Transfer the content from the current player to the bedroom player
- Transfer the content from the bedroom to the living room

During the test, we observed where the user was pointing during the drag-and-drop operations (on the device or on the representation of the device on the screen), and after we asked them to fill up a form. The form was divided in three parts: one about the prototype, one about the tangible drag-and-drop concept in general, and one with questions. The first and second part was questions with answers from 1 to 5, 1 for “not at all” and 5 for “definitely yes”.

About the prototype questions, it was to ensure that the quality of the experience was high enough to not interfere with the opinion about tangible drag-and-drop. Indeed, if a person did not like the experience, we want to know if the reason was the prototype or the concept. We asked if the remote control and the graphical user interface were easy to use, reactive and precise enough. All the answers are between 4 to 5, which is very good, except for the pointing precision of the Wiimote at the screen (some 3 and one 2, but an average of 4). Thus, the experience was good enough to fairly evaluate the tangible drag-and-drop concept.

The user evaluation of tangible drag-and-drop was encouraging as well. We asked five questions (check-boxes from 1 to 5) to users about the concept. The questions were, in this order:

1. Is this concept an interesting idea?
2. Are you convinced by this concept?
3. Is this concept natural?
4. Is this concept efficient?
5. Is this concept not demanding/tiring?

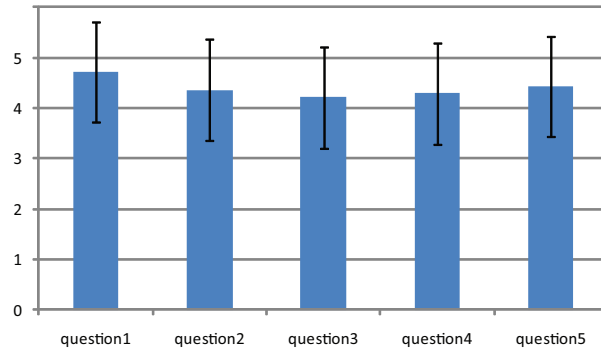


Fig. 4. User answers for the five questions about the concept. Columns represent the means of the 15 answers, the Y error bars are the standard deviations.

Results are presented in figure 4. As we can see, users were satisfied about all these points of the application.

Paradoxically, users pointed at the physical device only 57% of the time. This value moderates the natural aspect of the concept. After informally asking some questions to the users, several reasons can explain this result.

Firstly, it is important to consider the difficulty of fully breaking the border between digital and physical worlds. Indeed, all the participants were familiar with graphical user interface, and kept their focus on this interface in order to perform all possible actions.

Secondly, the feedback of the music device does not require to look at it. In [7], users looked at the photo frame to check if the picture transfer were done. In this audio prototype, users can “hear” if he successfully transfer the album, and for this reason can stay focused on the graphical interface.

Finally, the Fitt’s law[12] index of performance is inferior for tangible drag-and-drop than for the graphical user interface. Indeed, this value directly depends on the distance from the starting point to the center of the target, and the distance from the album selection interface to the digital representation of the device was smaller.

Despite that, it is important to take into account that the graphical interface was really uncluttered, with only three rooms, one device per room, and few options. With five devices or more on the GUI, or more than one device per room, the graphical interface becomes more complicated and the tangible drag-and-drop gains in interest. Some preliminary tests with two squeezebox in the same room are going in favor of this assumption.

In order to better assess the usability and applicability of the drag-and-drop interaction technique we performed a third experiment focused on evaluating the two variations of the technique: on-screen and spatial. Next subsection presents our results.

4.3 Evaluation of the Tangible Drag-and-Drop Technique

With this final evaluation, we wanted to compare separately what we call *on-screen* drag-and-drop technique -inside a graphical interface, and the *spatial* tangible drag-and-drop technique.

We created two applications out of prototype 2. In both applications, the concept was to play music, while sitting in a couch in the living room in front of the TV. The first application implemented an *on-screen* drag-and-drop interface (GUI-based). We kept the coverflow in order to display the albums, but we displayed only a picture of a Squeezebox device instead of the pictures of the rooms. If the user wanted to play an album, we had to drag-and-drop it from the coverflow to the device icon, and all the interaction regarding to the device (volume up/down, previous/next track, etc) was performed by pointing at this icon. The second interface was based on the tangible *-spatial-* drag-and-drop technique. The graphical interface on the TV was limited to the coverflow, and nothing else. Compared to prototype 2, we removed all the pictures of the rooms. If the user wanted to play an album, we had to drag-and-drop it from the TV to the device using the remote control.

For this study, we asked 10 participants (4 women, 6 men) to test our 2 interfaces, and to evaluate it with an AttrakDiff[6] questionnaire. All the participants were different from the previous tests. After a short training session, we asked them to play at least 4 albums with each system. After the test, we asked them their opinion concerning the two interfaces, and then to fill in the questionnaire. Figure 5 presents the results of the questionnaire.

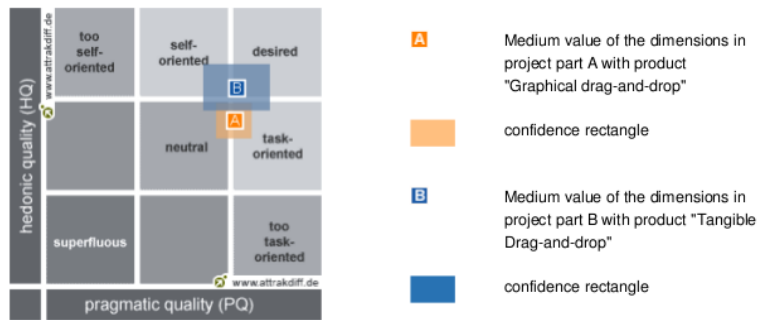


Fig. 5. Results from the AttrakDiff evaluation

From the AttrakDiff questionnaire, we can see that the tangible drag-and-drop technique performs better than the *on-screen* drag-and-drop technique. The difference in terms of pragmatic quality is not statistically significant, but the difference in terms of hedonic quality is important. The tangible drag-and-drop technique is perceived as much more human, innovative and attractive, but less predictable. The two interfaces are perceived as equally efficient, which explained the close result concerning the pragmatic quality.

During this evaluation, we learned that both interfaces are not mutually exclusive. Even if the interaction task - drag-and-drop - is the same, the two techniques are quite different.

Tangible drag-and-drop was appreciated because of its spatial aspect: you don't need to be precise, you don't even need to look at the device when you transfer an album on it. It is a natural way to interact when involved devices are within the field of view.

The graphical interface is more demanding in terms of precision: you have to be focused on the screen to interact with the system. The *on-screen* drag-and-drop was appreciated because of its iconic representation. You can interact with devices which are not in the field of view, like we did in the section 3.2. Another advantage is that you can create symbolic icons: for example, one user suggested implementing a Wikipedia icon that could be used to display the information of the album dropped on it. This kind of interaction is not possible using tangible drag-and-drop, as Wikipedia is not a physical interface.

To conclude this evaluation, we can say that the two interfaces are more complementary than in competition: one is using an iconic representation of the devices, the other one is more suitable for actions to be performed within a well-delimited area.

5 Conclusions and Future Work

Breaking the border between the digital and the physical world is difficult, especially for computer users who are used to stay focused on the graphical user interface in order to manipulate their digital content. Physical devices have to manifest their presence, to show that the tangible drag-and-drop action is possible when a particular content is selected. For example, a solution could be to illuminate the devices when the user is pressing the button. This way, it will attract her attention and the user will know that the drag-and-drop is possible on these devices. Another solution could be to add additional information directly on the remote control itself. Indeed, as users of the first prototype were complaining about the lack of feedback, we tried to add a picoprojector on the remote control. The idea is to display the selected content during the drag-and-drop action. There are two benefits of such an approach: besides providing visual feedback, it also helps to maintain the continuum between the virtual and the physical worlds. We conducted a small user experiment with this prototype, and these early tests are promising. Figure 6 is a photo of this prototype.

The action performed after a tangible drag-and-drop movement has to be clear and consistent. For example, if a user wants to visualize photos located on a digital camera, he/she will drag-and-drop the pictures from the camera to the television. But what will happen to the pictures located on the digital camera? Will they be erased or just transferred? Generally, it is better to divide the action as much as possible. If the user wants to erase the pictures, he will

perform another action (for example, drag-and-drop the pictures from the digital camera to a trash can). Another solution to inform the user about what will happen after a tangible drag-and-drop action could be to use a remote control equipped with a screen.



Fig. 6. Drag-and-drop remote control with pico projector

Tangible -spatial- drag-and-drop is interesting when all the devices are in range. It is interesting for example in the living room where there is often a large number of devices (television, VCR, media center, etc.). To transfer content physically from one room to another, a different paradigm such as on-screen drag-and-drop could fit better. Finally, one interesting point we have learned is that the on-screen drag-and-drop and the spatial drag-and-drop are not mutually exclusive, and could be used together to create a good user experience.

References

1. Alonso, M.B., Keyson, D.V.: MusicCube: making digital music tangible. In: CHI 2005 Extended Abstracts on Human Factors in Computing Systems (CHI 2005), pp. 1176–1179. ACM, New York (2005)
2. Edwards, W., Grinter, R.: At Home with Ubiquitous Computing: Seven Challenges. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) UbiComp 2001. LNCS, vol. 2201, pp. 256–272. Springer, Heidelberg (2001)
3. Enright, A.C.: Dissatisfaction Sows Innovation, <http://tinyurl.com/2zr7bw>
4. Fitzmaurice, G.W., Ishii, H., Buxton, W.A.S.: Bricks: laying the foundations for graspable user interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1995), pp. 442–449. ACM Press, New York (1995)
5. Graham, J., Hull, J.J.: I candy: a tangible user interface for itunes. In: CHI 2008 Extended Abstracts on Human Factors in Computing Systems (CHI 2008), pp. 2343–2348. ACM, New York (2008)

6. Hassenzahl, M., Burmester, M., Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualitt [AttracDiff: A questionnaire to measure perceived hedonic and pragmatic quality]. In: Ziegler, J., Szwillus, G. (eds.) *Mensch & Computer 2003. Interaktion in Bewegung*, pp. 187–196. B. G. Teubner, Stuttgart (2003)
7. Hopmann, M., Thalmann, D., Vexo, F.: Tangible Drag-and-Drop: Transferring Digital Content with a Remote Control. In: Chang, M., Kuo, R., Kinshuk, Chen, G.-D., Hirose, M. (eds.) *Learning by Playing. LNCS*, vol. 5670, pp. 306–315. Springer, Heidelberg (2009)
8. Hsu, S.H., Jumpertz, S., Cubaud, P.: A tangible interface for browsing digital photo collections. In: *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI 2008)*, pp. 31–32. ACM, New York (2008)
9. Ishii, H., Ullmer, B.: Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1997)*, pp. 234–241. ACM, New York (1997)
10. Lee, H., Jeong, H., Lee, J., Yeom, K.-W., Park, J.-H.: Gesture-Based Interface for Connection and Control of Multi-device in a Tabletop Display Environment. In: *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*, pp. 216–225. Springer, Heidelberg (2009)
11. Lee, H., Jeong, H., Lee, J., Yeom, K.-W., Shin, H.-J., Park, J.-H.: Select-and-point: a novel interface for multi-device connection and control based on simple hand gestures. In: *CHI 2008 Extended Abstracts on Human Factors in Computing Systems (CHI 2008)*, pp. 3357–3362. ACM, New York (2008)
12. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction* 7, 91–139 (1992)
13. Nunes, M., Greenberg, S., Neustaedter, C.: Sharing digital photographs in the home through physical mementos, souvenirs, and keepsakes. In: *Proceedings of the 7th ACM Conference on Designing Interactive Systems (DIS 2008)*, pp. 250–260. ACM, New York (2008)
14. Rekimoto, J.: Pick-and-drop: a direct manipulation technique for multiple computer environments. In: *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST 1997)*, pp. 31–39. ACM, New York (1997)
15. Rumbolt, R.B., McIntyre Jr., W.R.: Universal remote control unit. US Patent 4,774,511 (1988)
16. Seifried, T., Haller, M., Scott, S.D., Perteneder, F., Rendl, C., Sakamoto, D., Inami, M.: CRISTAL: a collaborative home media and device controller based on a multi-touch display. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS 2009)*, pp. 33–40. ACM, New York (2009)
17. Seifried, T., Rendl, C., Perteneder, F., Leitner, J., Haller, M., Sakamoto, D., Kato, J., Inami, M., Scott, S.D.: CRISTAL, control of remotely interfaced systems using touch-based actions in living spaces. In: *ACM SIGGRAPH 2009 Emerging Technologies (SIGGRAPH 2009)*, ACM, New York (2009) Article 6
18. Zigelbaum, J., Kumpf, A., Vazquez, A., Ishii, H.: Slurp: tangibility spatiality and an eyedropper. In: *CHI 2008 Extended Abstracts on Human Factors in Computing Systems (CHI 2008)*, pp. 2565–2574. ACM, New York (2008)