

a user interaction model for nfc enabled applications

Yaw Anokwa, Gaetano Borriello
University of Washington

Trevor Pering, Roy Want
Intel Research

1

Hi.

I'm Yaw Anokwa, a PhD student at the University of Washington.

This is work that was done in the summer 2006 with my advisor Gaetano Borriello in conjunction with Trevor Pering and Roy Want at Intel Research in Santa Clara.

The paper is called 'a user interaction model for nfc enabled applications' and you can follow along on page 357 of the proceedings, but be warned that work has evolved slightly.

First a quick outline.

outline

- ◎ NFC primer and motivation
- ◎ An interaction problem and previous work
- ◎ A potential solution and the contributions
- ◎ Future work and open questions

2

I've framed the talk so I can answer some of the questions we get about this work.

First, there will be a quick NFC primer and some of the motivation behind this work.

Second, I'll summarize what we feel is an interesting interaction problem in this space and what the previous work has accomplished.

Third, I'll discuss our solution and what the contributions are.

Finally, I'll talk about the work we have planned for the future and some of the open questions.

If you have questions, please interrupt at anytime.

nfc primer

- ◎ Near Field Communication (NFC)

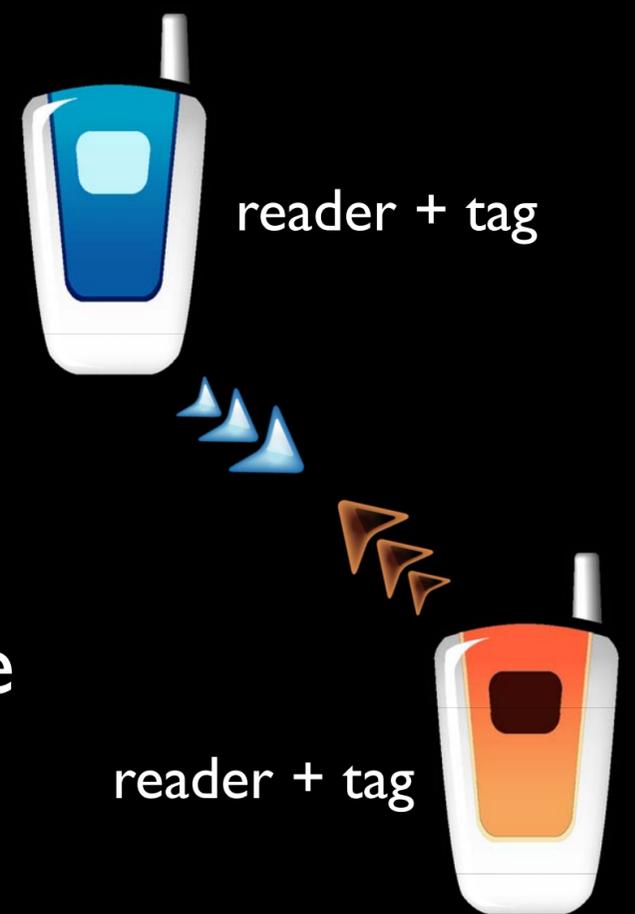
- ◎ Short range: 4 cm

- ◎ Max speed: 424 kbits/s

- ◎ NFC vs RFID

- ◎ Bidirectional transfer of data

- ◎ Mobile, low power, inexpensive



3

Most of you know this already, but just for review. What exactly NFC or Near Field Communication?

Let's say you have an RFID reader and a tag, and you stick it on a cellphone. Maybe get another a cellphone, and make it so the two phones only talk at extremely close range.

That is Near Field Communication.

NFC is a short-range RFID technology aimed for mobile devices. It works up to 4 cm (1.5") and the max speed is 424 kbit/s. In reality, it's a channel that is open when devices practically touch and is closed when devices move apart.

What makes NFC different from standard RFID is

- you can read/write tags, and transfer data between readers
- designed for mobile devices, low power, inexpensive

potential deployments

Get information by touching smart posters!



Your NFC device is your ticket!



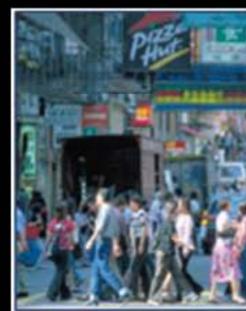
Your NFC device is your travel card!

NFC proponents imagine applications that vary from ticketing to travel



Buy goods from vending machines with your phone!

Get information about your current job or task!



Your NFC device is your credit card!

There are many potential deployments.

Applications are ticketing, purchasing, travel, getting bits of data, and device configuration. There are a lot of applications that NFC can be used for.

The aim here is that for each application it has to be simple and easy. This is important not only because that is NFC's goal, but it's because technologies that are not simple or intuitive are less likely to become a success.

potential deployments

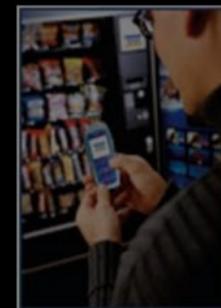
Get information
by touching
smart posters!



Your NFC device
is your ticket!



If NFC is to be successful,
we have to make sure we
keep it simple and easy



Buy goods from
vending machines
with your phone!

Your NFC
device is your
travel card!



Get information
about your current
job or task!



Your NFC device
is your credit card!

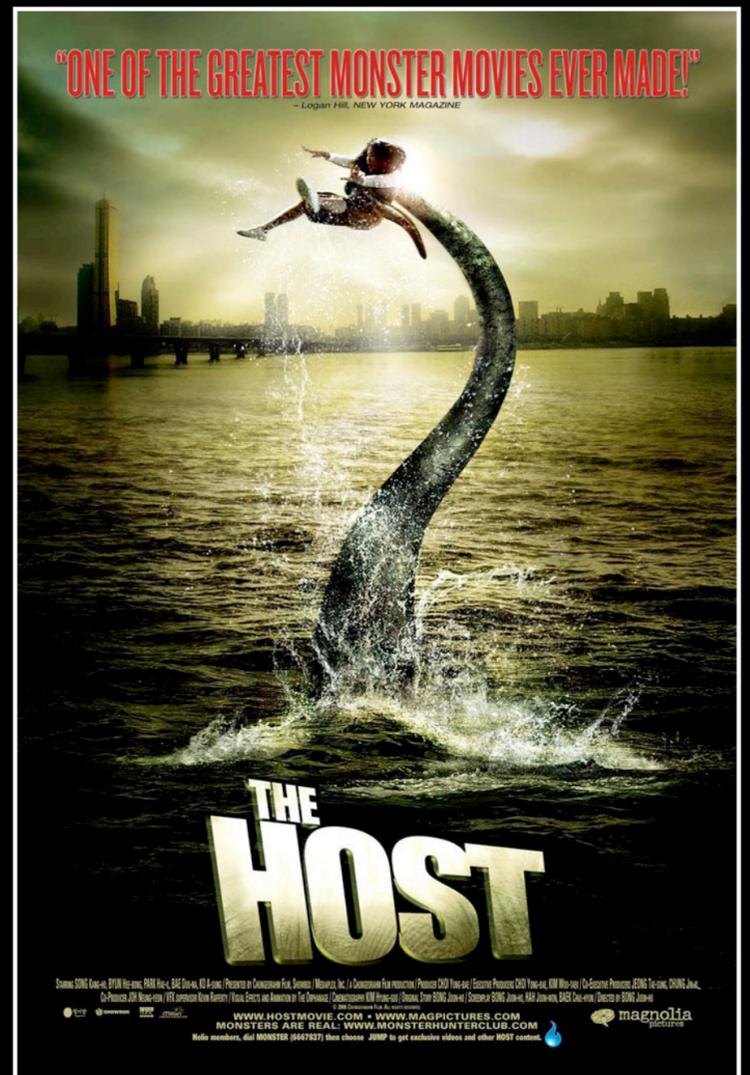
5

If we want NFC everywhere, we have to make sure it keeps its usability. We have to keep it simple and easy.

So let's pick one of these sample applications and run through a scenario.

purchasing a ticket

- ◎ Find smart poster
- ◎ Scan poster with phone
- ◎ Purchase ticket
- ◎ Go to theater
- ◎ Show ticket for movie
- ◎ Is it that simple?



We've all heard this scenario a hundred times. What if in the magical world of the future, I want to purchase a movie ticket from a smart NFC movie poster.

- I find a smart poster.
- I scan the poster with my phone.
- I purchase and get a ticket.
- I go to the theater, show the ticket and I'm done.

Or am I? Is it really that simple? What are the problems with this scenario?

purchasing a ticket

- ⦿ Read reviews about film
- ⦿ Watch trailers
- ⦿ Get directions to closest theater
- ⦿ Verify purchased ticket is correct
- ⦿ Give a ticket to a friend
- ⦿ Keep ticket stub as souvenir

7

Well the scenario ignores everything I could want to do at the poster! What if I don't want to buy a ticket? Can I still find this technology useful?

Maybe I want to read reviews about the film, watch a trailer or get directions to the closest theater.

Maybe, I when if I buy a ticket, I want to do the same things with it as a 'real ticket.' Can I verify the ticket is correct? Can I give the ticket to a friend as a gift, or can keep the ticket stub as a souvenir?

So you are now seeing what we feel is the problem.

the problem

- ◎ NFC is more than defaults, so knowing you have to scan something is not enough
- ◎ Just like we have 'WIMP' with GUIs, we need a model for NFC applications
- ◎ Solve the HCI problem of creating a model to mediate between the applications and data to make this a positive user experience

8

With NFC, no one has sat down to say this is how these apps will behave.

We argue that knowing you have to scan or touch is just not enough. Some applications are going to be complex, they may require multiple interactions. NFC gives us more tools to work with than touching to do a default action. We need to help people explore their spaces.

The parallel we like to draw here is with the personal computer. In the same way that the "window, icon, menu, pointer" model made the GUI successful, we need similar models for this class of applications.

So then the question is, can we solve the human-computer interaction problem and make this a positive user experience? Can we create a model of how we mediate between all the applications and data that we interact with?

We think we have a solution that may answer these questions, but first we want to go over some of the previous work.

related work

- ◎ Bridging Physical and Virtual [Want, 1999]
- ◎ Physical Browsing [Välkkyen, 2005]
- ◎ TagManager [Keränen, 2005]
- ◎ Experimental Comparison [Rukzio, 2006]

9

In 99, Want's idea was to derive some semantic meaning from the attached tag. For example, scanning an RFID enabled book could purchase it from an online book seller. This work changed the RFID space, but the automatic nature of the system means you couldn't explore other interactions.

Välkkyen define physical browsing as object selection when performed by either pointing, scanning or touching. They even build this on phones, but don't go into what happens after you point, scan or touch.

Keränen built a middleware component where users pair tags to a service so you define what each tag does. Which is a great idea, but we don't want the users setting up rules for every tag they encounter.

Rukzio conducted a user study evaluating the physical browsing idea and they find of the three (point,scan,touch) users heavily associate touch with security, speed, intuitive and error resistance. Clearly users know the power of touching.

So what our take on the problem?

model definitions

- ⦿ Items: NFC enabled device/tag (*poster*)
- ⦿ Objects: Things the item encapsulates (*tickets*)
- ⦿ Actions: Things objects can do or can be done to objects (*download trailer*)
- ⦿ Bags: Collections of objects and actions
- ⦿ Interactions: Using objects or doing actions
- ⦿ Events: Results of an interaction

10

First some definitions.

Items: Something NFC enabled. Say like a movie poster with a tag.

Objects: Things the item encapsulates or contains (including properties). A movie poster may have tickets, trailers, directions to a theater, etc.

Actions: Things objects can do and can be done to objects. So a trailer can be played, etc.

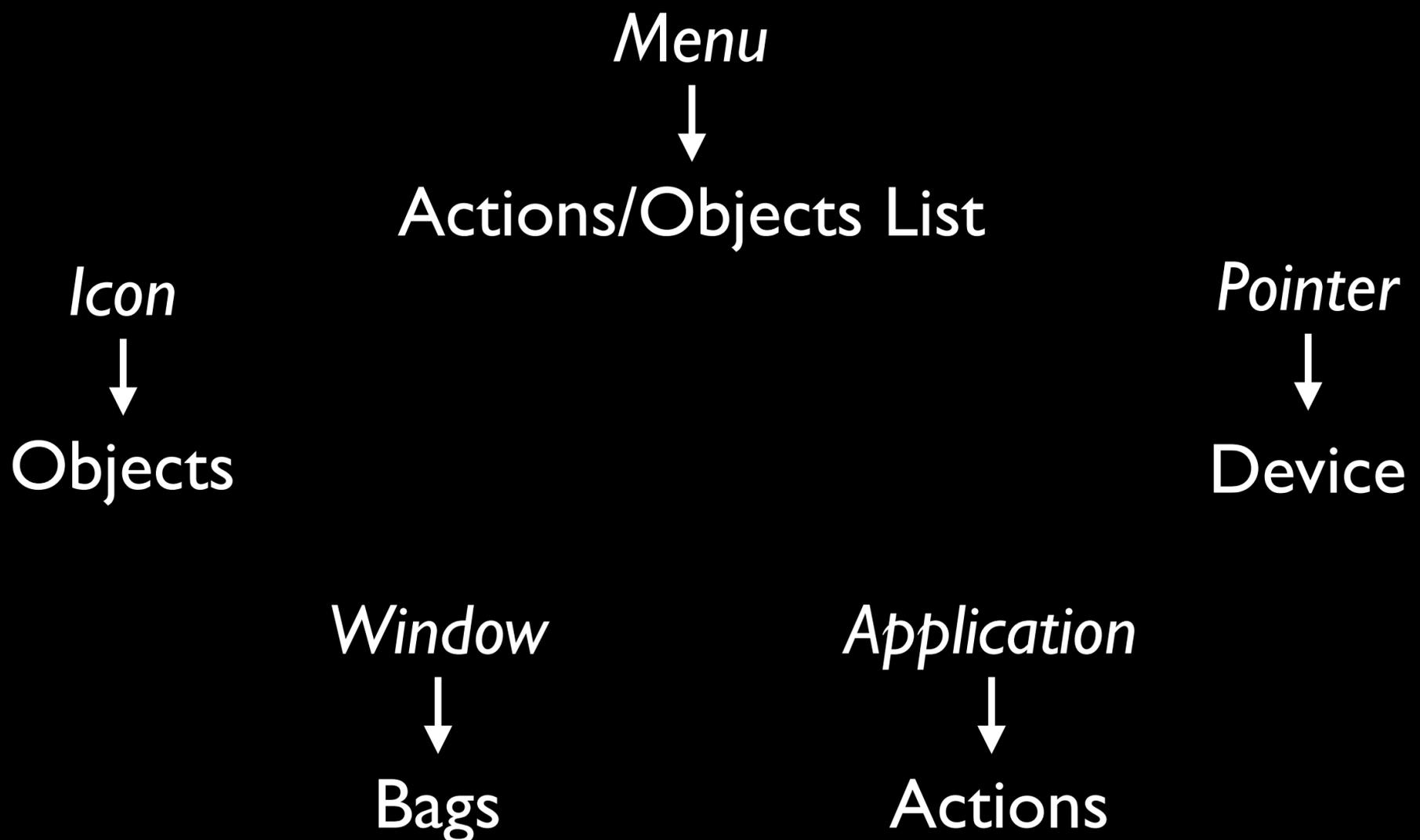
Bags: Collections of objects and actions. They are essentially a way to control a group of objects and actions.

Interactions: Using objects or doing actions.

Events: Results of an interaction

This should all be very natural and intuitive.

model definitions



11

There are similarities with the windows, icon, menu, pointer model here.

Menu is an object/action list

- List things you can do and can be done to objects

Pointer is the selection (on or off device)

- Selecting the item or object

Application is essentially a set of actions

- Things that can be done to object

Windows are bag

- Both are the things your work is constrained by

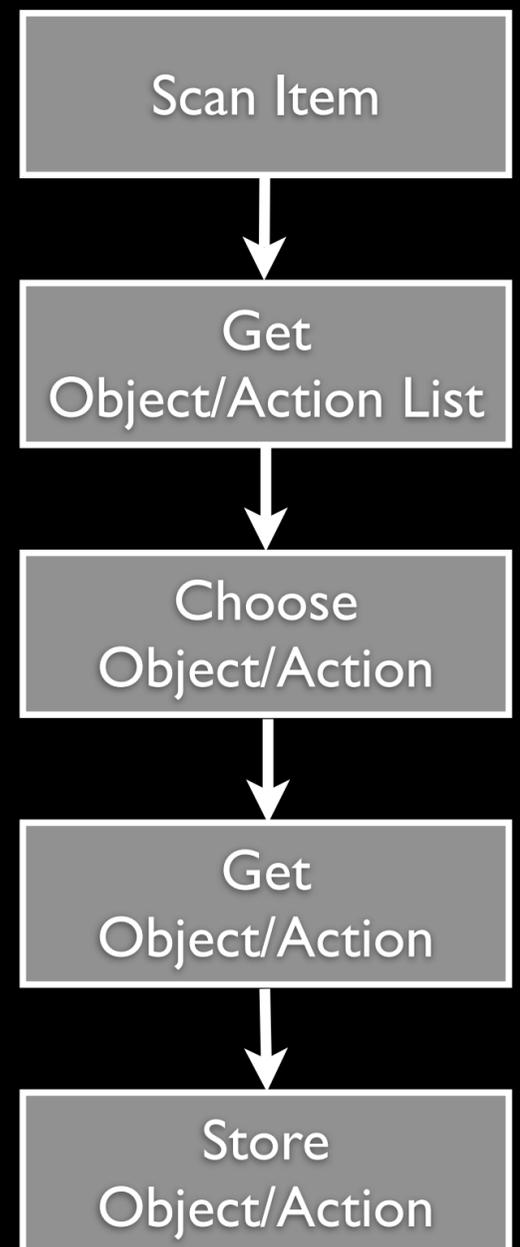
Icons are object

- Both are the things you work on

Again, it's not entirely 1-1 mapping, but it's there and it's a natural fit.

scanning an item

- ◎ Scan an item
- ◎ Get a listing of available objects and/or actions
- ◎ Select the objects/actions needed for interaction
- ◎ Transfer chosen objects/actions to device
- ◎ Store objects/actions in device bag



12

So how does the model work? Let's say I take my cellphone and I scan my item, a movie poster.

I get a listing of the objects and actions that are available. For example, I get a menu of the objects I can get (like a ticket) or actions I can do (like download a trailer).

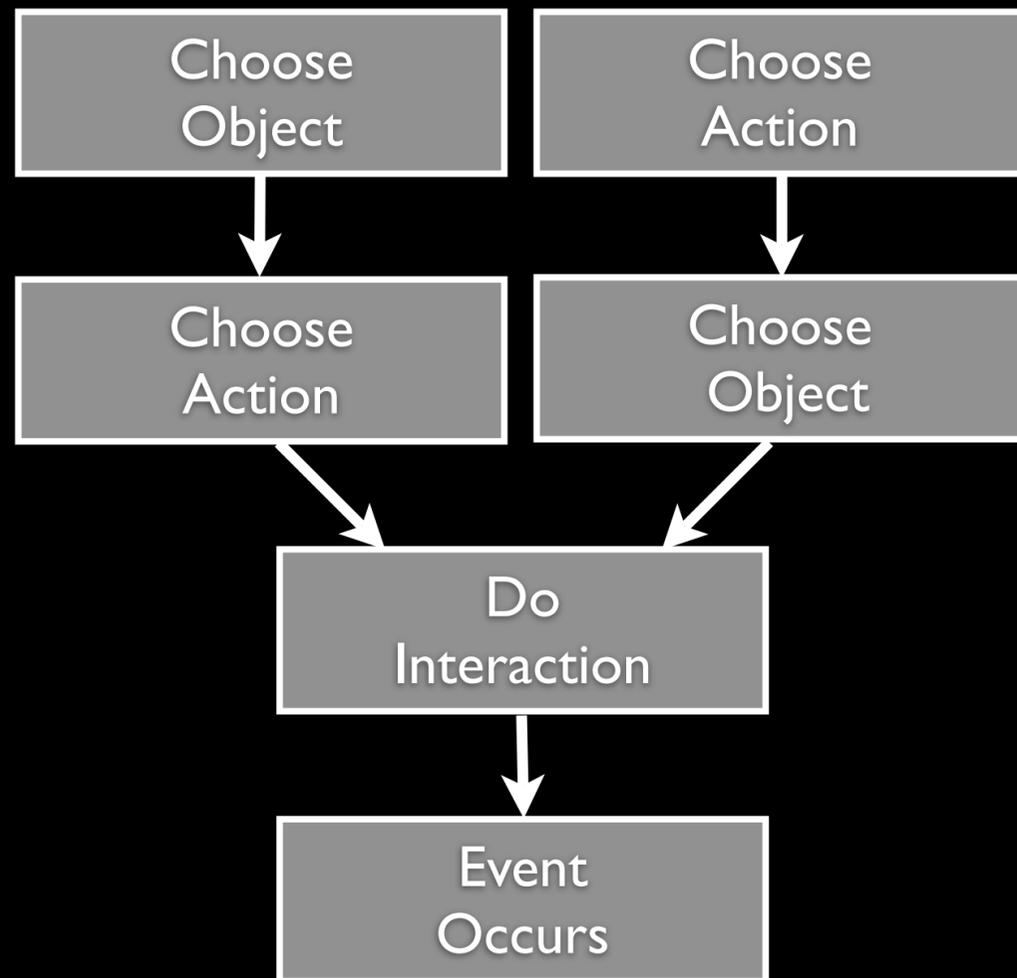
I then select the objects/actions I want. This can be specified by default or through a gesture or a manual selection.

Those objects/actions are transferred to your device. Once on device, are then stored in the device bag.

In this last two stages, the storage doesn't have to be permanent. I could do the object/action and make it transient. These stages can all be set to automatically run through these stages without confirmations.

using objects/actions

- ⦿ Choose an object/action
- ⦿ Choose an action/object
- ⦿ Choose action or object by itself from device bag
- ⦿ Do interaction
- ⦿ Event occurs



13

Once the scan is done, I can retrieve the objects/actions as needed.

I can either choose an object and pair it with an action. Say select a ticket and do the send to friend action.

Or choose an action and pair it with an object. Say play media then select the trailer.

I can also select just an object or action and rely on defaults.

Whichever you choose, you can do an interaction. This could be a gesture or some function on the device or even another touch.

The event then occurs. So again this is pretty natural model especially with defaults.

So what happens in our old scenario?

purchasing a ticket

- ⦿ Read reviews about film
- ⦿ Watch trailer now or get a reminder bookmark
- ⦿ Get and store directions to closest theater
- ⦿ Verify purchased ticket is correct
- ⦿ Give a ticket to a friend over NFC
- ⦿ Keep ticket stub as souvenir

14

When you scan the poster, you get a listing of all the things you can do. You can get information about reviews, trailers, direction and the like.

If you choose to purchase a ticket, it will show up as an object. You have an actionable item on your device you can use for other interactions. You can verify, send it to friends (over NFC), and keep remnants on the system.

Our model enables a lot of the things you would want to do, in a way that users can understand.

So to verify these claims, we've been building some hardware.

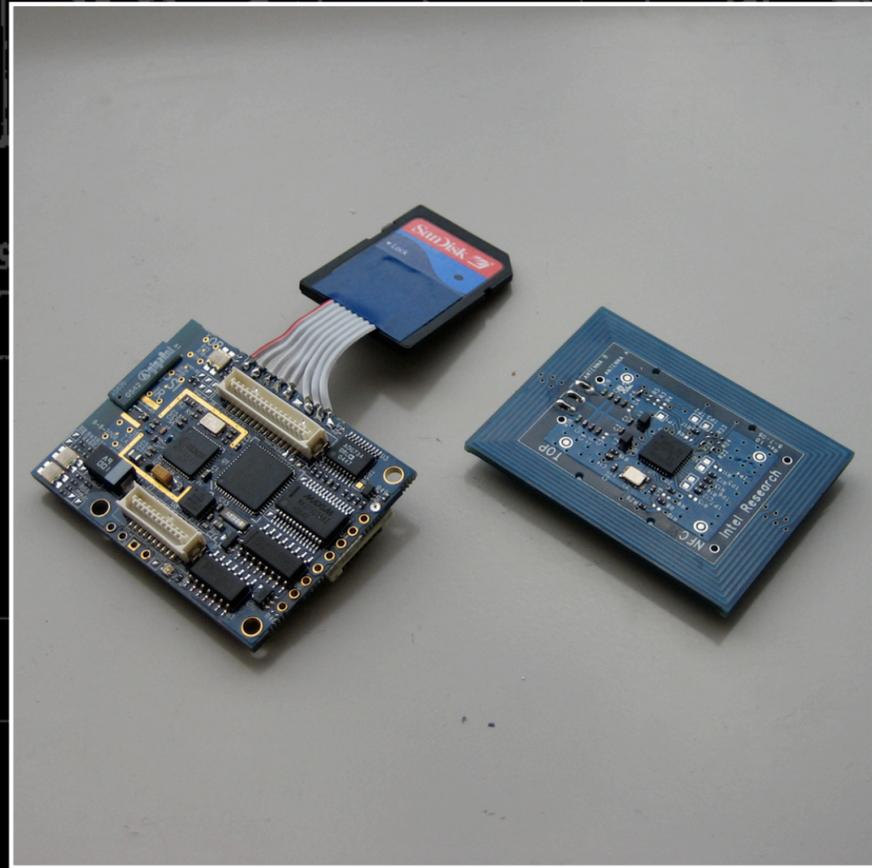
current work

- ⦿ Existing solutions are not open to experimentation
- ⦿ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ⦿ NFC Board: chip, antenna
- ⦿ System interfaces to SD slot
- ⦿ Demo app enables touch and gesture capability

Why build custom hardware? Existing solutions tend to be slow, don't have built-in Bluetooth/WiFi, or accelerometers. We wanted something open to experimentation to exploit the technology.

current work

- ⦿ Existing solutions are not open to experimentation
- ⦿ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ⦿ NFC Board: chip, antenna
- ⦿ System interfaces to SD slot
- ⦿ Demo app enables touch and gesture capability

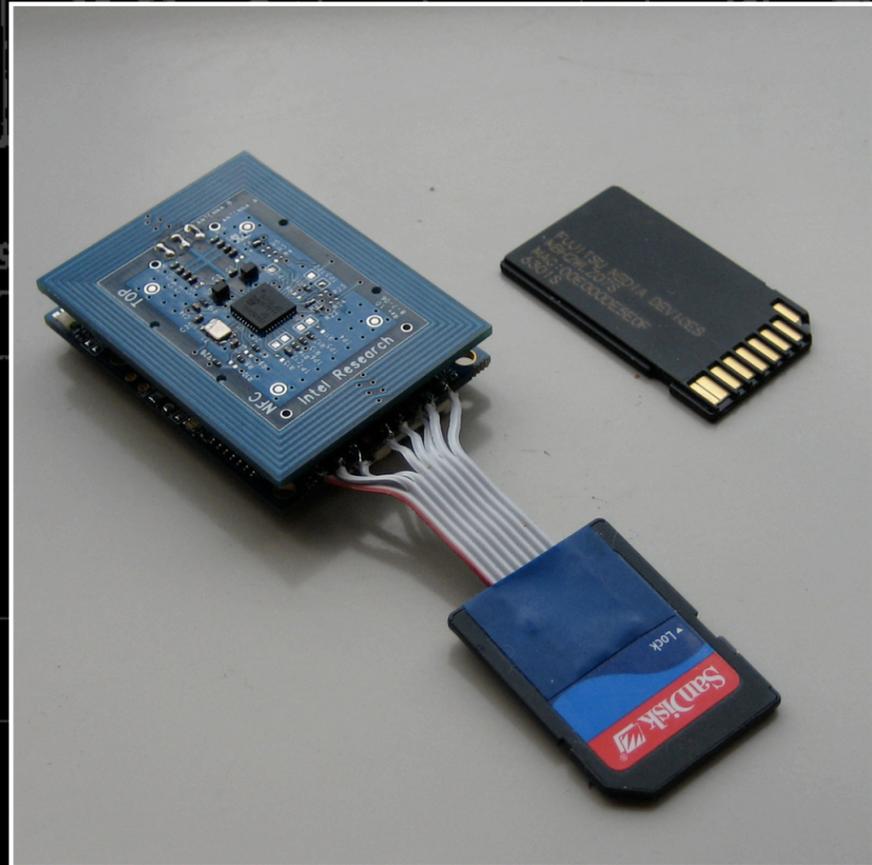


We have two boards. The one on the left is the phone system interface (PSI). Has a micro-controller, 802.14 radio, an accelerometer, and an SD slot.

The one on the right is the NFC board with antenna.

current work

- ◎ Existing solutions are not open to experimentation
- ◎ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ◎ NFC Board: chip, antenna
- ◎ System interfaces to SD slot
- ◎ Demo app enables touch and gesture capability



17

The NFC board sits on top of the PSI board and feeds into a standard SD slot. It's all multiplexed so you can use an SD card while doing the NFC interaction.

The electronics aren't that complicated -- you could fit it into the same shape as a WiFi dongle (on the right).

current work

- ⦿ Existing solutions are not open to experimentation
- ⦿ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ⦿ NFC Board: chip, antenna
- ⦿ System interfaces to SD slot
- ⦿ Demo app enables touch and gesture capability



When you put it in a Motorola Linux phone, it looks like this.

current work

- ⦿ Existing solutions are not open to experimentation
- ⦿ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ⦿ NFC Board: chip, antenna
- ⦿ System interfaces to SD slot
- ⦿ Demo app enables touch and gesture capability



We wrap it around the back and it fits nicely in your hand.

current work

- ⦿ Existing solutions are not open to experimentation
- ⦿ PSI Board: controller, 802.14 radio, accelerometer, SD slot
- ⦿ NFC Board: chip, antenna
- ⦿ System interfaces to SD slot
- ⦿ Demo app enables touch and gesture capability



Our demo app enables touch and gesture. The idea is that you touch two phones, exchange a list of objects and actions. You can then select what you want, and gesture to start a transfer of data or do an action.

contributions

- ◎ Our aim is a natural interaction model which keeps the HCI issues in mind
- ◎ Tags should attempt to encapsulate all the properties of the items it represents
- ◎ Readers should enable some model of objects/ actions so users keep some sense of possibilities
- ◎ Result of an interaction is an actionable item

21

Our approach is different from previous work in what we are trying to accomplish. It's an attempt to make sure that stakeholders in this space keep the HCI issues in mind when they build applications and deploy these systems.

We are not claiming our model is THE solution but we feel it's a natural interaction model.

Tags should encapsulate all the properties of the item it represents -- not just an identifier.

Once the readers should enable some model of interaction because it gives users a sense of what they can and can't do, while retaining many of the real world properties of the things they scan.

Finally, the result of your interaction is an actionable item. Whenever you complete an interaction, you get an icon that you can use in another interaction.

We are still working in this space. Specifically...

future work

- ◎ Building more applications and scenarios
- ◎ User evaluation to test the idea
- ◎ Gathering feedback about holes in model

22

We are building more apps and scenarios to test out try both mobile–mobile ideas and mobile–infrastructure ideas.

We currently setting up a user evaluation to see how users like this way of thinking about NFC.

Gathering feedback about the holes we have

- What happens when you have multiple devices?
- What happens when items contain many objects/actions?
- How do users to recover from mistakes?

Those are some of the questions we are thinking about. And so with that, are they any questions out in the audience?

questions?

web: <http://cs.washington.edu/homes/yanokwa/>