# Huddle: Automatically Generating Interfaces for Systems of Multiple Connected Appliances

*Jeffrey Nichols, Brandon Rothrock, Duen Horng Chau, Brad A. Myers*
Human Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{jeffreyn, rothrock, dchau, bam}@cs.cmu.edu
http://www.cs.cmu.edu/~jeffreyn/huddle/

## ABSTRACT
Systems of connected appliances, such as home theaters and presentation rooms, are becoming commonplace in our homes and workplaces. These systems are often difficult to use, in part because users must determine how to split the tasks they wish to perform into sub-tasks for each appliance and then find the particular functions of each appliance to complete their sub-tasks. This paper describes Huddle, a new system that automatically generates task-based interfaces for a system of multiple appliances based on models of the content flow within the multi-appliance system.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

**General terms:** Design, Human Factors

**Keywords:** Automatic interface generation, aggregate user interfaces, handheld computers, personal digital assistants, mobile phones, home theater, appliances, personal universal controller (PUC), Pebbles

## INTRODUCTION
The computerized appliances in our homes and offices are increasingly becoming connected, which allows multiple appliances to work together as a system to accomplish tasks that might otherwise have been impossible. Such connected systems are already commonplace for home theaters, presentation rooms, and video-conferencing systems.

The problem with many of these systems of appliances is that while their functionality is integrated, their user interfaces are not. In order to use these appliances together, users must learn to operate each appliance separately and also learn how each appliance's functions integrate with the others. This can lead to problems where the user correctly configures some appliances but not all. For example,

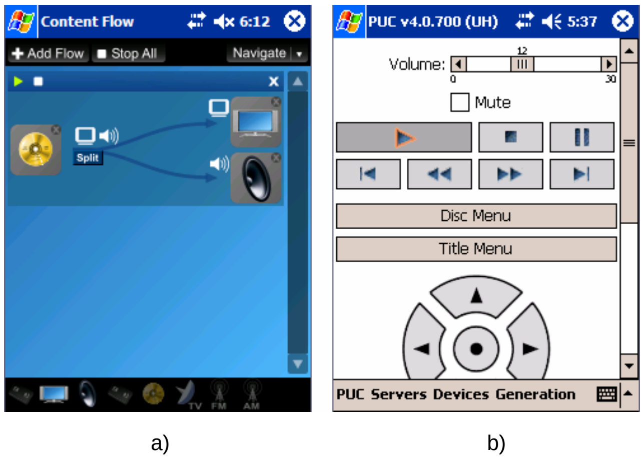


Figure 1. Interfaces generated by Huddle for a home theater system that is currently being used to watch a DVD through a television and audio receiver. a) The Flow-Based Interface (FBI) that allows users to specify their high-level goals, and b) an aggregate interface generated for the most commonly used functions of the content flow from the DVD to television and receiver.

a common home theater problem is seeing video from the desired DVD on the television but not hearing the sound from that same DVD on the system's speakers.

Today, some of these problems can be addressed by universal remote controls that integrate the user interfaces of each appliance into a single remote control device. These devices typically require a substantial amount of tedious programming however, such as entering the IR control codes for each button on each remote control, specifying sets of codes that must be sent to start each task, and designing panels of controls for use during each task. Many high-end users now pay professional system integrators to do this programming rather than try to do it themselves.

In this paper we present *Huddle*, a system for automatically generating integrated task-based user interfaces for a system of multiple connected appliances (see Figure 1). A key challenge for Huddle is to generate these interfaces without requiring substantial programming that is specific to each system of appliances. To address this challenge, Huddle relies on a description of content flow within each multi-appliance system. Content flows have two important properties, which make them ideal for our use:

1) Content flows seem to be closely related to user goals with multi-appliance systems. For example, in a home theater, the user may want to watch a DVD movie, which involves seeing the video on the television and hearing the audio through the stereo's speakers. To accomplish this, each of the appliances in the home theater must be configured to allow content from the DVD player to flow to the appropriate places.
2) The content flows of a system can be described as the separate flows within each appliance combined with a wiring diagram showing how all of the appliances are connected. This is an important property, because it divides the modeling work among the manufacturers of the appliances. The only system-specific input needed by Huddle is a diagram showing how the appliances are connected, which can be supplied by another application, a future wiring technology or the user.

Huddle uses knowledge of the appliances' functions and how these functions relate to the content flows to automatically generate a useful set of interfaces, and automatically configure appliances for any set of content flows. Huddle has three main features:

- A flow-based user interface that allows users to quickly specify a content flow. This interface allows the user to specify the endpoints for a flow, and optionally the path that the content should take between the endpoints (if there are multiple choices).
- A planner, based on the GraphPlan algorithm [2], that is able to automatically configure appliances to enable the user's desired content flows. If one or more requested flows cannot be activated, a question-answer interface can help the user fix the problem.
- *Aggregate interface generators*, which produce user interfaces that combine the functionality of multiple appliances based on the current set of active content flows and other parameters. The aggregate interfaces that we currently generate are: active flow controls, active flow setup, general setup, and merged controls.

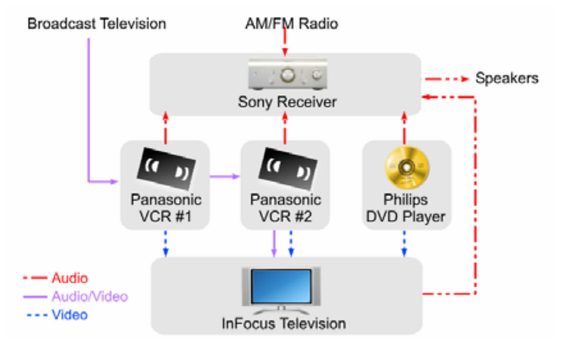
Huddle is implemented on top of our Personal Universal Controller (PUC) system [8], which previously generated remote control interfaces only for individual appliances. Like the PUC, users of Huddle use a handheld device such as a PDA or mobile phone to control all of the appliances in their multi-appliance systems. Huddle receives from each appliance an abstract specification of that appliance's functions, which also includes a description of the appliance's physical ports and internal content flows. Unlike other languages, such as UPnP, this specification does *not* describe how the user interface should appear. Huddle uses the specifications from each appliance and a description of how the appliances are connected to build a complete model of the content flow for the entire system of appliances. Huddle automatically generates user interfaces from this model.

In this paper, we start by putting Huddle in context with the related work in this area. Then we describe the scenarios that we will use as examples throughout the rest of the paper. In the next section we describe Huddle's architecture, followed by a discussion of our content flow modeling language. The following sections describe the user interfaces that we have created using content flows, including our flow-based interface with the integrated planner and the automatically generated aggregate interfaces. We conclude with a discussion of these techniques and future work.

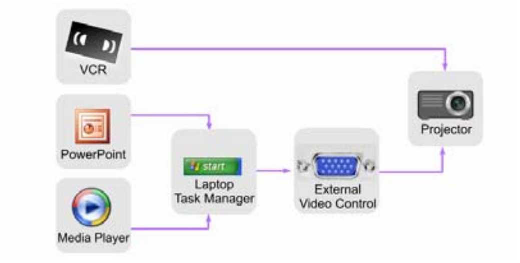
## RELATED WORK

For years many companies have been selling so-called "universal remote controls," which integrate the control of televisions, VCRs, and stereos into a single remote control unit. The problem with these remote controls is that the programming must be done manually, which can be a tedious and time-consuming task, especially for a large number of appliances. The most interesting of the devices in this class is the Harmony remote [5] from Logitech, which attempts to internally maintain a record of the current state for all of the appliances that it can control. This has the limitation that the remote must know the state of the system when it is first used and that all control must be performed using the Harmony remote afterwards, but, like Huddle, it has the advantage that the remote can hide functionality that is not available in the current state. The user interface is further simplified using a task-based approach that allows the user to select such options as "play movie in VCR" or "play DVD." The built-in set of tasks in the Harmony device is limited to the ones that the company has pre-programmed, and adding support for new tasks is nearly as laborious as programming a regular universal remote control. The pre-programmed tasks are a subset of the content flows that Huddle can support with its flow-based interface.

The Roadie system [4] provides a goal-oriented user interface for consumer electronics that may combine features of multiple appliances. Like Huddle, Roadie uses a planning algorithm to automatically configure appliances to match user goals. Unlike Huddle, Roadie uses a database of commonsense knowledge to find and understand possible user goals within the system. The user can specify the action they wish to perform using natural language, and then Roadie will attempt to interpret this action using its database and create a plan. Because the possible actions are restricted to the contents of the commonsense database, Roadie may not be able to support uncommon actions, such as those related to an uncommon configuration of appliances or to a new class of appliance that has just been added to the system. Huddle, in contrast, is able to acquire a model of the system from the appliances themselves, and thus is not subject to these limitations.



a) Home Theater



b) Presentation Room

Figure 2. Configuration of appliances in our home theater (a) and presentation room (b) scenarios.

The Stanford ICrafter [11] is a framework for distributing and composing appliance interfaces for many different controlling devices. To support composition, ICrafter relies on a set of "service interfaces" (that abstract the functionality of services) and a set of interface aggregators that are each hand-coded to build an interface for a particular pattern of service interfaces. When a user requests an interface for multiple services, ICrafter looks for an aggregator that matches the pattern and, if an aggregator is found, returns a single interface generated by that aggregator. For example, a camera might implement the DataProducer interface and a printer might implement the DataConsumer interface. The generic aggregator for the DataProducer/DataConsumer combination could then generate a combined interface for the camera and printer.

ICrafter's approach has several limitations however, which Huddle overcomes. First, a generic aggregator in ICrafter is only able to generate an interface for the common properties and functions shared by its service interfaces and none of the unique functions that may be implemented by a specific service. Second, ICrafter's generic aggregators are not able to include any design conventions that might be specific to the services. For example, a play button would be appropriate if a DataProducer was a DVD player but not if the producer was a camera. For ICrafter to produce interfaces with unique functions and appropriate design conventions, a special purpose interface aggregator would need to be built for the specific appliances involved. In contrast, Huddle's interface aggregation is faithful to the specific appliance interfaces that are being aggregated and includes all functionality of the connected appliances.

Omojokun et al. [10] have examined how remote control interfaces for a system of appliances might be generated from recordings of actual usage. Remote control usage was recorded by an IR receiver that was placed near the home theaters of several users, and the information gathered by the IR receiver was used to generate button-only interfaces. Huddle's generated interfaces are more sophisticated, but do not currently take into account previous usage. Including usage in Huddle's designs is an area for future work.

When designing Huddle, we considered the use of task models as an alternative to the content flow model that Huddle currently uses. Task models have been used as the basis for generating interfaces in many systems, such as TERESA [6]. We chose not to use task models in Huddle because we could not find a means to build a system-wide task model from independent pieces of a task model given by each appliance. There would be advantages to using task models however, such as improved description of the user's goals, and we will continue to consider the use of task models in Huddle in the future.

Several systems have explored the infrastructure issues that are involved in connecting and configuring systems of multiple appliances. One such system is Speakeasy [7], which uses mobile code to allow arbitrary devices and services to interact, and also to distribute user interfaces to the hand-held devices from which users interact. While Speakeasy might be able to automatically provide a wiring diagram to Huddle, it does not provide support for automatically generating user interfaces or for combining user interfaces for multiple appliances into a single aggregate user interface.

## MULTI-APPLIANCE SYSTEM SCENARIOS

In this section, we describe two scenarios, a home theater and a presentation room, that will be used throughout the rest of the paper to demonstrate the features of Huddle.

The home theater setup (see Figure 2a) includes five appliances: an InFocus television, a Sony audio receiver with attached speakers, a Philips DVD player, and two identical Panasonic VCRs. This setup supports many common tasks, such as watching television, watching a movie from either a DVD or videotape, and listening to the radio. It also supports a number of more complicated tasks, such as copying from a tape in VCR #1 to a tape in VCR #2, or watching television on one channel while recording up to two other channels. Sometimes tasks can be mixed, such as watching a sporting event on television while listening to a radio broadcast of the play-by-play. Certain tasks are impossible with this setup, such as recording a DVD to videotape, recording the radio, or recording from a tape in VCR #2 to a tape in VCR#1. As we will show below, Huddle's flow-based interface makes it clear to the user which flows are not possible.

The presentation room configuration has three physical devices: a projector, a VCR, and a laptop. The laptop's functions however, have been separated into several independent "logical appliances" which include the PowerPoint and Windows Media Player applications, the task manager, and control of the external video port. This configuration supports common presentation tasks such as showing slides, showing video from the laptop, and showing video from a VCR tape.

## ARCHITECTURE

In the Huddle system, a handheld device, such as a PDA or mobile phone, is the center for all communication with appliances and all interactions with the user. Huddle does not prevent users from interacting directly with the appliances however, and will not be disturbed if a user chooses to do so. An overall view of Huddle's architecture is shown in Figure 3.

Huddle requires three types of input in order to function. First, it requires a wiring diagram that describes how the multi-appliance system is wired together. Currently this diagram is specified by hand in XML, though we imagine that future wiring technologies might be able to automatically detect and provide this information. The wiring diagram contains a number of wire <begin, end> pairs corresponding to the physical wires that connect the appliances.

The second type of input Huddle requires is a PUC appliance specification from each of the appliances in the multi-appliance system. In addition to the description of the functions of each appliance from our previous work [8], Huddle requires the specifications include descriptions of the physical ports on each appliance and the internal content flows within each appliance. By combining this information together, Huddle creates a complete model of the possible content flows through the entire system (see the center portion of Figure 3). Huddle then uses this information to generate user interfaces.

Huddle also requires a knowledge base of functional similarity information, which is provided by our Uniform system [9]. This information allows Huddle to find functions with similar purposes across appliances in the system, which can be used to create interfaces that organize functions from multiple appliances in a meaningful way.

Huddle produces two kinds of interfaces to help users interact with their multi-appliance systems. The Flow-Based Interface (FBI) allows the user to quickly create and activate content flows between appliances by tapping or dragging the icons for desired sources and sinks onto the screen. The goal of this interface is to make high-level tasks easy to execute with the multi-appliance system.

Huddle also generates Aggregate User Interfaces (AUIs) that combine functions from multiple appliances into a single user interface. Various types of AUIs support different tasks within the multi-appliance system. The *Active Flow Control* AUI combines the most common control functions associated with the active content flows into a single interface, with the goal of making common content manipulations (such as volume control) easy to access. The *Setup* AUIs make infrequently used configuration parameters easy for the user to access, with the goal of supporting expert usage of the appliance system. Finally, AUIs merge some functions that occur on multiple appliances into a

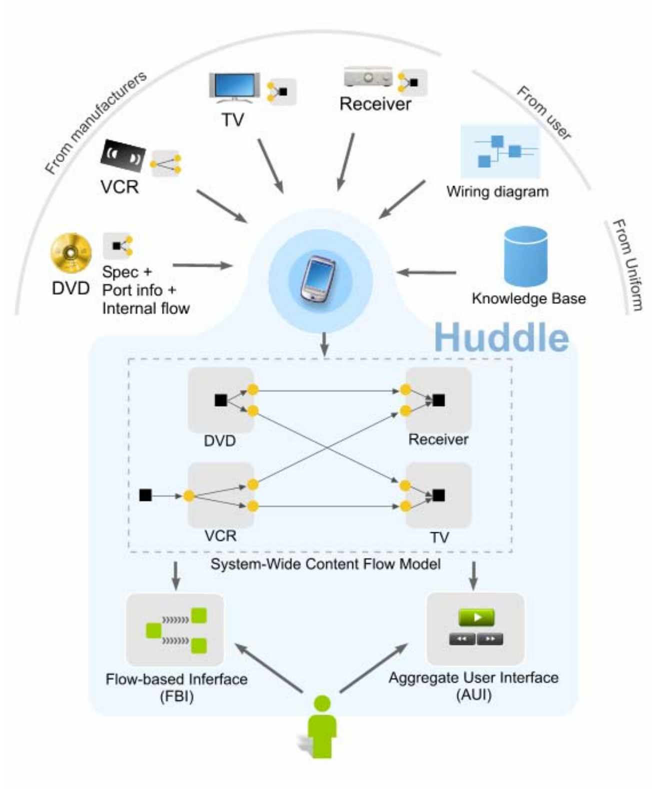


Figure 3. Architecture of the Huddle system.

single point of control on one interface. This allows the user to do such things as set the current time in our AUI and have this change automatically broadcast to each appliance in the system.

## CONTENT FLOW MODELING

Huddle adds two new sections to the PUC specification language to specify the physical ports of the appliance and the internal content flows that use those ports. An example from the Philips DVD Player for both of these sections is shown in Figure 4.

An early question in this project was whether Huddle's description of the input and output ports of the appliance should match the physical ports of the appliance or be abstracted in a way that made the content flows convenient to specify. We chose an intermediate approach that represents all of the physical ports but also includes port-groups that allow the content flow specification to refer to multiple physical ports using a single label. We hope that including the physical ports in the specification will allow us to build a usable configuration tool in the future that can give specific wiring instructions to the user. This tool could help the user wire up their system while at the same time providing Huddle with the required wiring information.

The content flows within an appliance are represented using three different structures:

- **Sources** represent content that originates within the appliance, such as from a DVD player playing a DVD or a VCR playing a videotape. Display devices that have internal tuners, such as televisions receiving broadcast signals through antennas, are not defined as sources however, because the content does not originate inside of the tuning device. Instead, broadcast signals are described as a special "external source" that must be routed through a tuner to be viewable by the user.
- **Sinks** represent locations where content may either be displayed to the user or stored for later retrieval. For example, the television screen, receiver speakers, and VCR tape (for recording) may all be sinks for content in our home theater scenario.
- **Pass-throughs** represent an appliance's ability to take in some content as an input and redirect it through one or more of its outputs. For example, the InFocus television in our scenario has the capability of taking the audio it receives as an input and making it available as an output for other appliances. Tuning appliances, such as cable television set-top boxes, are also represented as pass-throughs, which usually take a multi-channel input from an antenna and output single channel data.

The pass-through structure is particularly important, because it allows Huddle to track the flow of content from its origination point, through multiple appliances, to its final destination. Previous systems, such as a Speakeasy [7] and Ligature [3], have used only sources and sinks to model the path of data within a system. Using their approach, it is difficult to know whether the content a device is receiving as input is being redirected through an output, which makes determining the full content flow impossible. Without knowledge of the full content flow from start to finish, we could not infer the task that the user is trying to perform and generate a useful interface for it.

Each content flow of an appliance will usually have a dependency formula that specifies when that content flow is active based on the state of the appliance. For example, in Figure 4 the DVD player's source is active when the DVD player is powered on, a disc is in the player, and the play mode is not stopped (stop corresponds to a value of "1" in this specification). Content flows are also explicitly linked to the physical ports of the appliance, and each of these ports may also have a dependency formula that specifies when that particular port is being used by the content flow. For example, in Figure 4, the S-Video output of the DVD player is only available when the DVD player's progressive scan feature is turned off. This dependency information is used by Huddle to automatically determine how to activate a particular flow of content based on the current state of the multi-appliance system.

Channels are an important concept in the Huddle content modeling language. When a pass-through or sink receives a multi-channel input, a channel variable may be specified from the appliance that specifies the particular channel being tuned. The content flow language can also specify that one channel of a multi-channel stream is being replaced by the appliance, which is often used by VCRs to specify that the output of the tape source can appear on channel 3 or 4.

```
<ports>
  <outputs>
    <port-group name="SD-Output"
                           content-type="video">
      <port name="Video" content-type="video"
                           physical-type="RCA"/>
       <port name="S-Video" content-type="video"
                            physical-type="S-Video"/>
    </port-group>
    <port-group name="Progressive Scan Output"
                            content-type="video">
      ...
    </port-group>
    ...
  </outputs>
</ports>
<content-flow>
  <source name="Disc" content-type="av">
    <active-if>
      <equals state="Base.Power">
        <constant value="true"/></equals>
      <equals state="Base.Status.DiscIn">
        <constant value="true"/></equals>
      <not>
        <equals state="Base...PlayControls.Mode">
          <constant value="1"/></equals>
      </not>
    </active-if>
    <output-ports>
       <port name="SD-Output.Video">
        <active-if>
          <equals state="Base...ProgressiveScan">
            <constant value="false"/></equals>
        </active-if>
      </port>
      ...
      <port-group name="Audio">
        <active-if>
          <equals state="Base...AudioMute">
            <constant value="false"/></equals>
        </active-if>
      </port-group>
      ...
      </output-ports>
      <objects>
        <group name="Base.Controls.Common"/>
        <group name="Base.Setup.Audio"/>
        <group name="Base.Setup.Video"/>
      </objects>
    </source>
```

Figure 4. Example of the physical port and content flow model from our Philips DVD Player specification

Each content flow and each of the ports to which it links may also specify a set of objects and groups within the appliance specification that are related to the appliance's processing of the content in the flow. This is useful, for example, for specifying that the volume control on the receiver is related to the speaker sink, or that the tint, contrast, and brightness controls are related to the television's screen sink. The information about related objects is very important for generating Huddle's aggregate user interfaces.

## FLOW-BASED INTERFACE

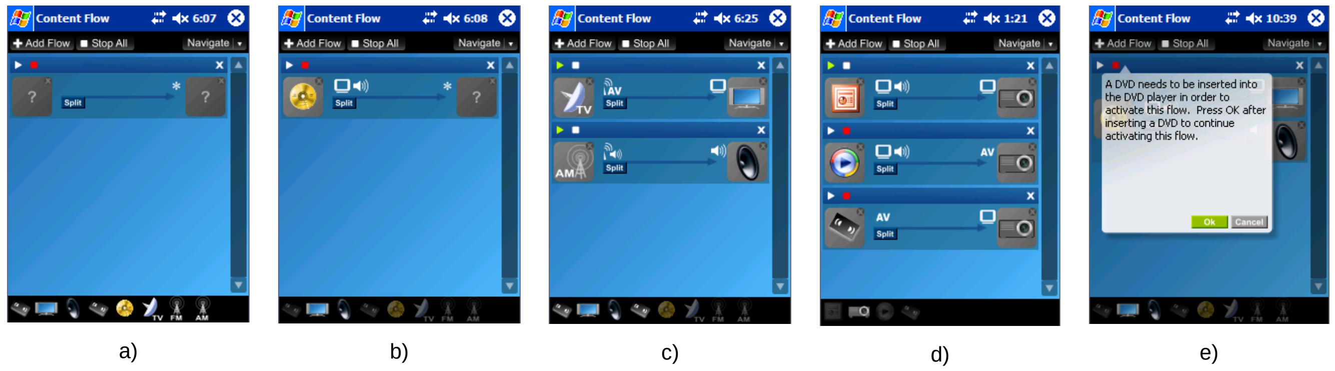
The Flow-Based Interface (FBI) is designed to allow users to quickly specify a flow from one source of content to one or more content sinks. For example, the user might specify a flow from the DVD Player's disc to the television's screen and the receiver's speakers. When the user activates this flow, Huddle inspects the dependencies of each of the flow's elements, generates a plan to satisfy these dependencies, and executes that plan to enable the flow. If the flow cannot be enabled, perhaps because of other active flows that the user has already specified, the system will prompt the user with a dialog box and attempt to help the user resolve the problem. Several examples of the FBI in action are shown in Figure 1a and Figure 5.

To make the idea behind the FBI clear, we will describe the interaction that a user would have with the interface in order to start watching a DVD movie. Figure 5a shows the FBI in its initial blank state. Near the top of the screen is a blank flow with empty spaces for a source and sink, with an arrow between them. At the bottom of the screen is the appliance bar, which contains an icon corresponding to each appliance that has a source or sink in the system. Additional flows may be added to the screen, by pressing the "Add Flow" button at the top of the screen, and the scrollbar on the right allows for scrolling when more flows have been added to the list than can be shown. There are two usage scenarios that we envision for this interface: the user creates flows for each of her common tasks and switches among them, or the user uses just one flow and modifies it as necessary to suit the current task.

When a user wishes to begin a flow, she drags an icon from the appliance bar to one of the empty spaces in the blank flow. The empty space highlights when she drags the icon over it, indicating that the appliance icon may be placed there. Once the DVD icon has been placed in the source location (see Figure 5b), content type icons appear on left side of the arrow and the icons in the appliance bar corresponding to appliances that cannot be sinks for the DVD player source are grayed out. This includes icons that correspond only to sources, such as the broadcast television and radio icons, and the VCR icons that cannot be sinks for DVD content because of our home theater's particular wiring configuration. The user can now see that the receiver and the television are the only available appliances that will work with the DVD player. In this scenario, the user first drags the television to the empty sink space on the flow. At this point, the green Go button will become enabled because this configuration corresponds to a valid flow (Note that the television speakers can be a sink for audio content). The asterisk above the arrow on the right side indicates that the flow-based interface will infer the type of content to route to the television based on the specified sinks.

The user now wishes to add the receiver as an additional sink. To do so, she presses the "Split" button underneath the arrow on the left. This causes the flow to be split into two arrows and a new empty sink space to be created. To add the receiver's speakers as a sink, the user can then drag the receiver to the empty space. The result is the view shown in Figure 1a. In this scenario, Huddle is able to automatically infer that the TV speakers should not be used, because an audio sink was added to the flow. If the user wanted audio to come from both sets of speakers, she could indicate this by tapping the content type icon next to the television and selecting the audio/video content type.

The user can now click the play icon in the flow's title bar, which invokes the planner to automatically activate this flow. If a successful plan is found, the appliances will be automatically configured, the play icon will turn green, the stop icon will turn white, and a bubble will appear to inform the user that the flow has been activated. If a plan cannot be found, a bubble will appear to help the user resolve the problem (see Figure 5e). One difficulty with planning algorithms, such as the GraphPlan algorithm that Huddle uses, is that they cannot produce useful error messages when planning fails. Therefore, Huddle first uses two



A DVD needs to be inserted into the DVD player in order to activate this flow. Press OK after inserting a DVD to continue activating this flow.

Figure 5. Screenshots of the Flow-Based Interface in various states of use. a) The blank interface before any sources or sinks have been dragged onto the interface. b) The interface after the DVD player has been added. Note that only the available sink appliances, the television and the receiver, are available in the appliance bar at the bottom of the screen. c) The interface being used to watch a sporting event on television but listening to the play-by-play over the radio. d) The interface being used during a presentation to manage the display of different sources, with a PowerPoint slideshow as the current source. e) A dialog bubble from the question/answer interface for resolving planning conflicts.

approximation checks to search for conflicting appliance variables and active flows, allowing us to produce a more useful error message.

Our first conflict check searches the dependencies of the newly specified flow to see if any read-only variables have values that make activating the new flow impossible. Such variables usually reflect the physical status of the appliance, which the user can address once informed. For example, the DiscIn variable of the DVD player might be set to false when the user pressed the play icon in our previous example. If this happened, the system would then ask the user if they can rectify this problem. Although the current language for this error message can be somewhat stilted, we do provide predefined strings for common problems, such as there being no disc in the DVD player (see Figure 5e).

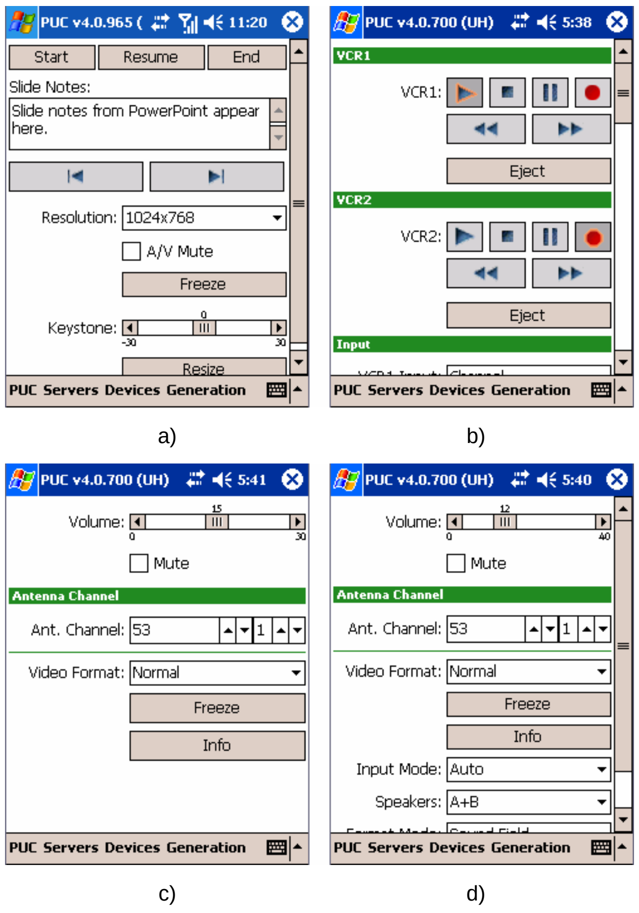
After we check for variable conflicts, we check to see if any currently active flows conflict with the new flow. To perform this check, we examine the dependency information associated with the newly specified flow and the dependencies for any active flows, looking for variables that must have more than one value for the flows to be active simultaneously. If this situation is found, then we can immediately go back to the user to ask which of the conflicting flows the user wants to use now.

Once we have found that no obvious conflicts exist, we execute the planning algorithm to find a valid plan for activating our new flow. If a plan is found, then the system will carry out that plan to create the right configuration of variables that will activate the new flow and maintain the state of any existing flows. The planning algorithm may still fail however, such as when second-order dependencies conflict. In our experience, these conflicts are rare, but when they occur we ask the user to choose between finding a plan that activates the specified flow and disables the currently active flows or finding a plan that activates the specified flow without considering the effects on other flows. If a plan is found in either case, we prompt the user again before carrying out the plan to make her aware of which flows will be deactivated by the new plan.

The flow-based interface also provides interaction to navigate to the other user interfaces provided by Huddle and the underlying PUC system. In the upper right corner of the FBI is a "Navigate" pull-down menu, which allows the user to navigate to the different aggregate user interfaces that Huddle can generate (discussed next). Double-clicking on any appliance icon, either in the appliance bar or in a flow, allows the user to navigate to the full interface for that individual appliance.

## AGGREGATE USER INTERFACES

The FBI provides an interface for users to accomplish high-level goals within the multi-appliance system, such "watching a DVD movie" with a home theater. There is still a need however to provide the user with finer-grain control of the individual appliance functions. For example, the user may wish to pause the DVD while it is playing to take a phone call, or go to the next slide in a PowerPoint presentation. A user might also discover that the movie is too dark requiring adjustment of the brightness of the television, or that the keystone setting needs to be adjusted on the projector.



Slide Notes: Slide notes from PowerPoint appear here.
Resolution: 1024x768
A/V Mute
Freeze
Keystone: 0 (-30 – 30)
Resize

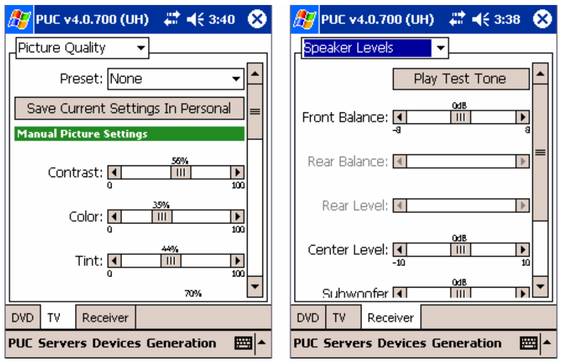
Volume: 15 (0 – 30)
Mute
Antenna Channel
Ant. Channel: 53 1
Video Format: Normal

Volume: 12 (0 – 40)
Mute
Antenna Channel
Ant. Channel: 53 1
Video Format: Normal
Input Mode: Auto
Speakers: A+B

Figure 6. Active Flow Controls AUIs generated for two different flows: a) PowerPoint being shown on a projector and, b) the VCR1 tape being recorded onto the VCR2 tape. c) Broadcast television content being viewed on the television's screen and using the television's internal speakers, and d) the same broadcast television content being viewed on the television but with the audio coming from the audio receiver's speakers.

To address these problems, Huddle provides the user with several Aggregate User Interfaces (AUIs) that combine functions from each of the appliances in the system to create useful task-specific interfaces. Huddle currently can generate four different AUIs: Active Flow Control, Active Flow Setup, General Setup, and Merged Functions.

It is important to note that the user also has access at any time to the full interfaces for each appliance that the PUC system is already able to generate. Thus it is not our goal to provide access to the full set of appliance functionality through the set of aggregate interfaces, but instead to provide interfaces to meaningful sets of functionality from the collection of appliances in the system.

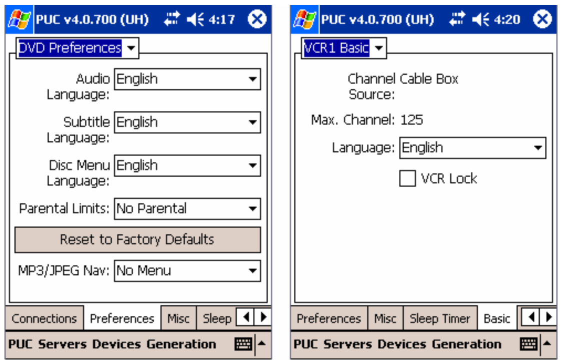


Preset: None
Contrast: 56% (0–100)
Color: 35% (0–100)
Tint: 44% (0–100)
70%

Front Balance: 0dB (-8–8)
Rear Balance:
Rear Level:
Center Level: 0dB (-10–10)
Subwoofer 0dB

a) b)

Figure 7. Two shots of the Active Flow Setup AUI for the DVD player to receiver and television flow. Note that the interface is organized by appliance, as shown by the tabs at the bottom of the screen.



Audio Language: English
Subtitle Language: English
Disc Menu Language: English
Parental Limits: No Parental
MP3/JPEG Nav: No Menu

Channel Source: Cable Box
Max. Channel: 125
Language: English
☐ VCR Lock

a) b)

Figure 8. Two shots of the General Setup AUI for our home theater setup. Note that in both shots, the tabs at the bottom of the screen represent high-level concepts within which the functions are organized by appliance (combo boxes at top).

### Active Flow Controls

The Active Flow Controls AUI combines commonly used functions that are related to the currently active flows. Figure 1b and Figure 6 show examples of active flow control aggregates generated when the active flow is playing a DVD to the receiver and television, controlling a slideshow in a presentation room, copying a tape from one VCR to another, and watching television with the audio coming from the television or receiver speakers.

Huddle identifies functions to be used in the Active Flow Controls AUI in two stages. In the first stage, functions are extracted from the appliance's specifications that are either mentioned in the flow dependencies for the currently active flows, or are noted as being related in the appliance's content flow model. In the second stage, these functions are filtered to select only the most common functions that users will likely want to manipulate. Huddle uses two heuristics in the filtering stage because no information is available in the PUC specification to directly identify the commonly used functions of an appliance.

The first heuristic is to eliminate any functions associated with "Setup." Nearly all specifications contain a high-level group with the name "Setup" or something similar. This group will often be identified by the Uniform system [9], which provides a knowledge base of similarity information for PUC specifications. Huddle uses Uniform's mapping information to identify the Setup group in each appliance and then filters out any functions that are contained in these Setup groups.

The second heuristic eliminates any functions that, if used or modified, would always cause the flow to stop being active. This eliminates all power functions (which can be easily accessed elsewhere), the input-select variables from stereo and television, the VCR/TV functions of the VCRs in some situations, and a number of other variables that may be common but would overlap with the functioning of the FBI. The exceptions to this rule are media control functions, such as play, stop, and pause, which are always included. Although the user may deactivate a flow by pressing stop or eject, we feel that users would be annoyed if these functions were not easily available and that users can easily recover if they use these functions in a way that deactivates a flow.

Once the set of functions has been decided, the functions are organized into a list. Functions from the source are placed first in this list, followed by functions from any sink appliances, with functions from any pass-through appliances at the end. We use this ordering because it seems that functions from the end points of a flow are usually the most relevant to the user. The exceptions to this rule are the volume and mute functions, which we automatically place at the top of the generated interface to ensure that they are easy for users to access. Any functions in this list found to be functionally similar, either as identified by Uniform or because they are the same function on a different instance of the same appliance, are then grouped together in the final interface. For example, this grouping caused the play controls for each of the VCRs to be located next to each other in Figure 6b.

Figure 6c and d illustrate two Active Flow Control aggregates for slightly different flows, watching broadcast television with using just the television and watching broadcast television via both the television and audio receiver's speakers. Note that these interfaces at the top are quite similar, although in Figure 6c the volume slider is controlling the television's volume function where the same slider in Figure 6d is controlling the receiver's volume.

### Active Flow Setup

The Active Flow Setup AUI combines setup functions that are related to the currently active flows. Huddle identifies the functions for this aggregate using the first stage of the process used for the Active Flow Controls AUI. Unlike for that earlier aggregate however, the second stage filtering process for this aggregate takes only functions that are found within the "Setup" group. This process typically finds functions that affect the output of the currently active flows but will be used infrequently, such as the brightness and contrast controls for the television and the speaker level controls for the receiver. This AUI will still not include any controls that cause the flow to stop being active, however, since those are best controlled using the FBI.

The Active Flow Setup AUI is organized by appliance, because we found that a desired setup function was typically easier to find with this organization. See Figure 5 for two shots of the Active Flow Setup interface generated for the flow for the DVD to receiver and television. We originally tried to organize this aggregate by content type, which in the case of a home theater would give top-level groups for audio and video, but this created lower quality interfaces. The approach worked reasonably for appliances whose functions could be classified by their place in the content flow, such as the receiver and television which in some flows receive only audio and video content respectively. However we found that for appliances which handled both audio and video content that this approach relied too much on Uniform's ability to identify sub-groups that corresponded to Audio and Video. It is possible that this approach might be viable with improvement to Uniform.

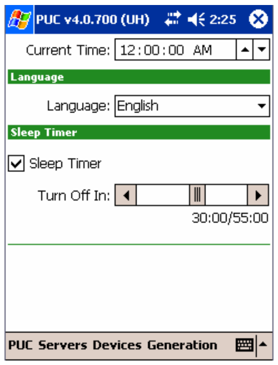
### General Setup

The General Setup AUI (see Figure 8) combines setup functions across all of the appliances that are not related to any content flow. These functions typically include things such as parental content restrictions, time functions, software upgrade controls, and the configuration of defaults. The functions for the AUI are extracted by iterating over the specifications for each of the appliances and eliminating all of the functions that could be used in the previous two AUIs. An additional filtering step removes any functions that are not in the "Setup" group.

The General Setup AUI is organized first by any high-level collections of functions that we can identify as existing on more than one appliance, and then by appliance. We first attempt to identify these high-level collections with Uniform but also search for any top-level groups within the Setup groups that may have the same name. Groups with the same name may not be identified by Uniform because their contents are not identified as being similar enough. These groups are often catch-all groups such as "Preferences," which have a large variance in the types of functions that they contain.

### Merged Functions

There are a few settings across a system of appliances where a single value should be applied to all appliances rather than laboriously setting the value on each appliance. Examples include the time on the clock, the language (e.g., English), and the sleep timer (that turns off the appliance after a selected number of minutes). Other settings that occur across appliances, however, should not be merged. For example, it is usually wrong to set the channel of the VCRs and television to the same value simultaneously or to set all the devices to be powered on at the same time. Even setup functions cannot always be combined depending on the particular function and how similar the functions are



Current Time: 12:00:00 AM

Language: English

☑ Sleep Timer

Turn Off In: 30:00/55:00

Figure 9. The Merged Function AUIs generated by Huddle featuring the clock, language, and sleep timer functions on a single panel.

across appliance types. For example, the DVD player and the television both have a contrast setting, but it would be inappropriate to set both of them simultaneously.

The Merged Functions AUI handles the small number of functions that are appropriate to combine (see Figure 9a). As with the previous aggregates, we use Uniform's knowledge base to identify similar functions across appliances.

As future work, we plan to explore how the volume function might be merged across appliances using the flat volume technique developed by Baudisch et al [1]. While this work has been shown to apply quite well to volume, it is unclear whether it would be applicable for other kinds of settings, such as brightness or contrast.

## DISCUSSION AND FUTURE WORK

The most important concept behind the design of Huddle is its use of a content flow model to help users accomplish their high-level goals. This approach seems to work well for the constrained domains of home theaters and presentation rooms, and we believe that it can be extended to support many more features than we have discussed here. For example, with more detailed modeling of content types, Huddle should not only be able to find content flows for users' goals, but also to find the optimal path for the particular content that the user is viewing. This is a particularly important problem as the types of content within the home theater grow to encompass numerous high-definition video and audio standards which may be supported at varying levels by different appliances and different types of wires.

A problem that we have been considering is how content flows can help Huddle understand that the lights need to be dimmed in order to view a projected PowerPoint presentation. An extreme solution would be to extend the content flows all the way to the final content sink at each user's eyes and ears, although this would require extensive modeling of each room, its lighting, and the user's perceptual capabilities in order to be successful. A more practical approach may be to provide basic models of which lights and projectors interact, perhaps with a few "environment content sinks" for the most important locations in a room that can allow Huddle to reason about the interactions of appliances within the environment.

We also believe that the content flow concept will prove extensible to other appliance domains, such as video-conferencing systems and even manufacturing processes. The extensibility of Huddle seems to be limited by the correspondence between the content flows and the tasks that the user wants to perform. Where there is correspondence, such as in the scenarios we have considered here, our approach works well. One scenario where Huddle may not be effective is in the kitchen, where tasks often center on recipes. It seems that many recipes use the same content flow through appliances, which suggests that the content flow may not be descriptive enough to generate useful task-based interfaces for the appliances.

There are two important problems of multi-appliance systems that Huddle does not currently address: helping with the initial wiring of the system and trouble-shooting problems when they occur. Both features could be added to Huddle using Roadie's [4] approach, which relies on a planning system similar to Huddle's. Some of the wiring problems could also be addressed in a tool that helps users specify the diagram that Huddle needs to build its system-wide content flow model. This tool could also help users determine how to best wire their system to support all the flows that they expect to use. It is worth noting that Roadie takes a different approach to configuration, by including wiring instructions in the plans that it generates for each user task as the user is using the system. This ensures that users are always able to perform a task if it is possible with some configuration of their system, but it seems better to perform this kind of analysis at setup time since, in our experience, it is unlikely that users will want to rewire their system on a regular basis.

Huddle currently generates four different kinds of aggregate user interfaces, and we plan to explore both improving our existing set and building new kinds of aggregate interfaces. A promising direction that we intend to explore is a usage-based aggregate interface, perhaps based on the ideas of Omojokun et al [10]. We are also interested in combining a usage-based aggregate with information about the user's context to provide the right function at the right time.

We are also planning to conduct a formal evaluation of Huddle's interfaces. We have already done some informal evaluations with prototypes of the Flow-Based Interface, which we used to improve the final version that is described in this paper. We are currently planning a user test in which we will put users in front of an existing home theater system and compare their interactions between the existing remote controls and Huddle.

A large part of the problem discussed in this paper arises from multiple appliances being connected together, which requires the user to interact with multiple interfaces to accomplish a single task. An obvious solution here is to integrate the appliances into a single monolithic appliance for which an interaction designer can carefully construct a good user interface. In fact, this solution can be seen for some consumer electronics, such as for shelf stereos which integrate an amplifier with a CD player, radio tuner, and other audio devices. The problem with this approach is that it does not allow for expandability and innovation. If all audio appliances had been integrated ten years ago, then today there would be no place for devices like the iPod. Huddle allows users to easily interact with systems of appliances, which enables appliance manufacturers to pursue the design of new appliances that may be added to these systems.

## CONCLUSIONS

Huddle demonstrates that powerful and general home theater interfaces can be created by combining interaction design, planning algorithms, and automatic interface generation techniques. This work demonstrates that automatic interface generation systems can extend the capabilities of human designers by adding features that would be difficult or impossible to implement by hand.

## ACKNOWLEDGMENTS

This work was conducted as a part of the Pebbles project, and was funded in part by grants from NSF, Microsoft, and General Motors. The National Science Foundation funded this work under Grant No. IIS-0534349. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

## REFERENCES

1. Baudisch, P., Pruitt, J., and Ball, S. Flat Volume Control: Improving Usability by Hiding the Volume Control Hierarchy in the User Interface, in *CHI'2004*: 255-262.
2. Blum, A. and Furst, M. Fast Planning Through Planning Graph Analysis, in *International Joint Conference on Artificial Intelligence (IJCAI 1997).* 1997: 1636-1642.
3. Foltz, M.A. Ligature: Gesture-Based Configuration of the E21 Intelligent Environment, in *MIT Student Oxygen Workshop.* 2001
4. Lieberman, H. and Espinosa, J. A Goal-Oriented Interface to Consumer Electronics using Planning and Commonsense Reasoning, in *Intelligent User Interfaces.* 2006: 226-233.
5. Logitech, "Harmony Remote Control Home Page," 2006. http://www.logitech.com/harmony/.
6. Mori, G., Paterno, F., and Santoro, C., Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. *IEEE Transactions on Software Engineering*, 2004. **30**(8): 1-14.
7. Newman, M.W., Izadi, S., Edwards, W.K., Sedivy, J.Z., and Smith, T.F. User Interfaces When and Where They are Needed: An Infrastructure for Recombinant Computing, in *UIST'2002*: 171-180.
8. Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., and Pignol, M. Generating Remote Control Interfaces for Complex Appliances, in *UIST'2002*: 161-170.
9. Nichols, J., Myers, B.A., and Rothrock, B. UNIFORM: Automatically Generating Consistent Remote Control User Interfaces, in *CHI'2006*: 611-620
10. Omojokun, O., Pierce, J.S., Isbell Jr., C.L., and Dewan, P., Comparing End-User and Intelligent Remote Control Interface Generation. *Personal and Ubiquitous Computing*, 2006. **10**(2-3): 136-143.
11. Ponnekanti, S.R., Lee, B., Fox, A., Hanrahan, P., T. Winograd. ICrafter: A service framework for ubiquitous computing environments, in *UBICOMP 2001*: 56-75.