MIDAS:

MULTI-DEVICE INTEGRATED DYNAMIC ACTIVITY SPACES

A Dissertation

by

UNMIL PURUSHOTTAM KARADKAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2011

Major Subject: Computer Science

MIDAS: Multi-device Integrated Dynamic Activity Spaces

MIDAS:

MULTI-DEVICE INTEGRATED DYNAMIC ACTIVITY SPACES

A Dissertation

by

UNMIL PURUSHOTTAM KARADKAR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Richard Furuta
Committee Members,	Frank Shipman
	John Leggett
	Lauren Cifuentes
Head of Department,	Duncan (Hank) Walker

December 2011

Major Subject: Computer Science

ABSTRACT

MIDAS: Multi-device Integrated Dynamic Activity Spaces. (December 2011)

Unmil Purushottam Karadkar, B.E., University of Pune; M.S., Texas A&M University

Chair of Advisory Committee: Dr. Richard Furuta

Mobile phones, tablet computers, laptops, desktops, and large screen displays are increasingly available to individuals for information access, often simultaneously. Dominant content access protocols, such as HTTP/1.1, do not take advantage of this device multiplicity and support information access from single devices only. Changing devices means restarting an information session. Using devices in conjunction with each other poses several challenges, which include the presentation of content on devices with diverse form factors and propagation of the content changes across these devices. In this dissertation, I report on the design and implementation of MIDAS—architecture and a prototype system for multi-device presentations. I propose a framework, called 12C, for characterizing multi-device systems and evaluate MIDAS within this framework.

MIDAS is designed as a middleware that can work with multiple client-server architectures, such as the Web and context-aware Trellis, a non-Web hypertext system. It presents information content simultaneously on devices with diverse characteristics without requiring sensor-enhanced environments. The system adapts content elements for optimal presentation on the target device while also striving to retain fidelity with the original form from a human perceptual perspective. MIDAS reconfigures its presentation in response to user actions, availability of devices, and environmental context, such as a user's location or the time of day.

I conducted a pilot study that explored human perception of similarity when image attributes such as size and color depth are modified in the process of presenting images on different devices. The results indicated that users tend to prefer scaling of images to color-depth reduction but gray scaling of images is preferable to either

modification. Not all images scale equally gracefully; those dominated by natural elements or manmade structures scale exceptionally well. Images that depict recognizable human faces or textual elements should be scaled only to an extent that these features retain their integrity.

Attributes of the 12C framework describe aspects of multi-device systems that include infrastructure, presentation, interaction, interface, and security. Based on these criteria, MIDAS is a flexible infrastructure, which lends itself to several content distribution and interaction strategies by separating client- and server-side configuration.

To Flora, Eva, Meena, and Sushila—the women in my life.

ACKNOWLEDGEMENTS

This dissertation is undoubtedly my most significant tangible artifact that will remind me of my days in Aggieland. It is supported strongly by several friendships, life experiences, hobbies, and memories that have shaped me.

For a student town the extent to which Aggieland is dominated by family is always surprising to me. Living away from my own family taught me to appreciate it more. Years of separation from Aai, baba, Saurabh, Manmath, and Akka aji has led me to experience that blood is thicker than anything in the world. I cherish every moment that I get to spend with my beloved Eva and our daughter, Flora.

In addition to relatives, I have been lucky to experience “the Aggie family” firsthand. The pachucos are a family that has slowly spread all around the world. Luis and his family, Kelly and MOK, Ebby and Jessi, Avital and Rohit, Brent, and Bryan started out as friends but have transcended that descriptor a long time ago.

The key aspect of a successful career is mentorship. I have been blessed to learn from the wisdom, insight, and temperament of Dr. Furuta. Dr. Shipman’s guidance on a variety of matters has shaped my worldview and also research. Dr. Leggett’s vision in shaping the research agenda for institutions and communities is awe-inspiring. Dr. Cifuentes has been crucial in channeling my thoughts about the role of user evaluation.

Much of my involvement in extra-curricular activities is a direct result of the sense of civic duty instilled in me by Curt Carver and Daniel Ragsdale. You guys taught me to contribute back to the community we live and work in.

MIDAS extends caT. Work by several fellow-students has provided critical interface with caT. In particular, Tolga Ciftci implemented a socket-based communication interface. Jeevan Joseph John developed the Web site used for conducting the study regarding image attribute similarity. Young Joo Park implemented the Resource Manager. Yungah Park and Jin-Cheon Na lent me their experience and helped troubleshoot aspects of caT-MIDAS communication.

Friends, such as Haowei, Michael, Luis, Avital, and Pratik have enriched my years in the CSDL. The theological and philosophical discussions with Neal and Paul added a different hue to typical technological conversations I often heard in the department.

To one and all, who made my time in Aggieland an unforgettable one, thank you! I am eternally in your debt, whether I have specifically mentioned you or not.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Over the past decade the computing and information access infrastructure available to users has diversified considerably. On one hand, mobile devices are competing for dominance against desktop and notebook computers, which served as primary means of Web access in the 1990's, as they have become increasingly powerful and popular in the new century. Improvements along four axes can be attributed to the quick adoption of these devices: mobile phone networks, application development platforms, content adaptation, and physical characteristics. Simultaneously, higher-end computing platforms that include large displays and novel interaction modalities, such as multi-touch displays, are also becoming commonplace. This increasing heterogeneity of the computing infrastructure has resulted in individuals possessing multiple mobile computing appliances and having access to large-screen displays in social spaces. However, typical Internet protocols, such as HTTP, restrict users to one computer at a time and require them to restart their activity should they switch clients. My research explores mechanisms that enable users to jointly harness the characteristics of all their appliances for a richer information access environment. In this dissertation, I report on the design and development of Multi-device Integrated Dynamic Activity Spaces (MIDAS), a software architecture that supports users in flexible, simultaneous, multi-computer information access. MIDAS users can interact with multiple devices simultaneously and their actions, such as following a link, may result in changes to information elements rendered on other computers as well. MIDAS presentations are flexible as they can reconfigure document distribution dynamically if computers become unavailable or new ones become available. For example, the battery of a mobile phone may discharge, leaving it inoperable or a tabletop display may be within a user's reach as she walks into a conference room.

This dissertation follows the style of *ACM Transactions on Information Systems*.

1.1 BACKGROUND

Wide advent of 3G and, more recently, Long Term Evolution (LTE, commercially branded as 4G [InfoWorld 2011]) networks for mobile access have made it possible for users to connect to the Web from almost anywhere at speeds comparable to those of wired computer networks [Sprint 2011; Verizon 2011]. LTE networks are targeting data transfer speeds in excess of 1 Gbps in low mobility settings, for example while walking [Mogensen et al. 2009]. Thus, handheld computers are quickly becoming a viable platform for high-bandwidth information access.

"Mobile device" is no longer a monolithic classification. These appliances come in several flavors and form factors, including smart phones (iPhone, Droid, and HTC Evo), tablet computers (iPad, Galaxy Tab), and e-readers (Kindle, Nook). Each category consists of models that vary in display size and resolution. For example, while a typical smart phone screen is about 3" at a resolution of 320X480 pixels, some older smart phones only include a 2", 160X196 pixel display, while the HTC Evo boasts a 4.3", 480X800 pixel screen. The display of the first generation Kindle measures 6"; the Kindle DX has a 9.7" display and the Nook, 3.5". While e-ink-based readers typically support only 4 bit-gray-scale displays, LCD-based displays can display vivid 24-bit color.

The Cascading Style Sheets (CSS) 2.1 specification expects the client browser to select one of the pre-specified styles via the media type [Freeman and Freeman 2006; Bos et al. 2011], allowing designers to present renditions optimized for smart phones, desktop computers, or printers. Updates to this standard recognize that pre-specified media types supported by CSS 2.1 can no longer generate effective renditions on all mobile computers. Media queries included in the CSS3 specification support dynamic content adaptation by querying for the characteristics of a presentation environment [Lie et al. 2011]: for example, display resolution, device orientation, and browser window size [Hogan 2010].

The displays of desktop and notebook computers are diversifying as well. Notebooks and netbooks with displays between 7" and 17" are typical. Displays for desktop computers typically range between 17” and 30”. Large digital television sets with displays up to 56” are Internet-capable and hence, viable for Web access. Furthermore, surface computers, such as Microsoft’s Surface [Microsoft 2011a] and PQ Labs’ G3 multi-touch tables [PQ Labs 2011] are becoming increasingly affordable. Microsoft has partnered with Sheraton Hotels [Microsoft 2008] and AT&T [Perez 2008] to set up its tables in public areas such as hotel lobbies and mobile phone retail stores.

Major manufacturers are increasingly using new display technologies, such as organic light-emitting diodes (OLEDs), in television and mobile phone screens [Howard 2004]. The electroluminescent molecules that constitute the OLEDs result in thinner, lighter, flexible, and power-efficient screens when compared to the traditional liquid crystal displays (LCDs) that dominate today’s devices. OLEDs can also be printed using inkjet technology on a variety of non-rigid surfaces. This property makes them tremendously attractive for manufacturing personalized, flexible displays in custom sizes. The K-12 edition of the New Media Consortium’s 2010 Horizon report projects that such displays will be adopted for use in textbooks and other mass media in four to five years [Johnson et al. 2010]. Use of flexible, custom-made displays will cause the form factors and display properties to diversify even further and place several screens within reach of individuals and groups.

Users already possess multiple information appliances, for example smart phones, tablets, and notebook computers, and have access to others, such as desktop computers, printers, and projected displays, in their offices and homes. Yet, typical users find few, if any, applications that let them use these devices together when accessing information, especially Web-based information. The networking capabilities of mobile devices are improving rapidly and, while we are used to accessing the Web from mobile as well as home or office settings, the existing Web communication protocols do not support users in transitioning seamlessly between these settings. Changing devices means having to restart their activity from scratch. To take advantage of device multiplicity and diversity, access protocols should support concurrent use of multiple appliances, providing an information experience unhindered by the boundaries and limitations of individual devices.

1.2 PROBLEM STATEMENT

This research aims to support a user in accessing digital information in a heterogeneous, multi-device environment. While the limitations of one device may make it unsuitable for presenting certain elements (for example, large images or PDF files cannot be displayed effectively on a mobile phone), another could render these without compromising quality. Thus, the first goal is to design a software architecture that will deliver documents to multiple appliances simultaneously.

In addition to documents, interactive elements will also be directed to multiple computers. The software architecture must be able to respond to user input received from any of the active appliances. For example, when a user, who is browsing simultaneously from a tablet and a mobile phone, clicks a link on the phone the display on the tablet may change as well. While the appliances should all work together, my second design goal is that this cooperation should not require sensors embedded in the user's environment as such settings are still hard to find. Unlike the computers outlined in Weiser's vision of the 21st century computer [Weiser 1991], MIDAS aims to rely only on widely supported technologies, such as wired and wireless Internet connections.

To account for the rapidly increasing heterogeneity, this architecture should not make assumptions about the characteristics of the connecting devices. It is possible that no available appliance may be able to present information elements in their original form. My third design goal is that architecture must be able to transform such elements for delivery in a variety of contexts. For example, if a user only has access to a smart phone, a large textual document may be summarized for a quick preview or may be converted to audio form. A large image may be scaled, cropped, color-reduced, or only its caption may be displayed. When such transformations are made, the system must retain the integrity of the original form from a human perspective to whatever extent possible, which is my fourth design goal. My fifth design goal is to present each document on the device that is best capable of rendering it.

Finally, the appliances within reach of a user may change during a session. A smart phone may run out of battery or a mobile user may sit down at a table to start her notebook computer. The sixth design goal is to reconfigure information renditions in response to device availability (or unavailability) in order to render information by employing the best characteristics among all devices within a user's purview. MIDAS is designed to meet these goals.

1.3 SCENARIO

Eva, a student enrolled in a freshman art survey class, is a MIDAS user. She is

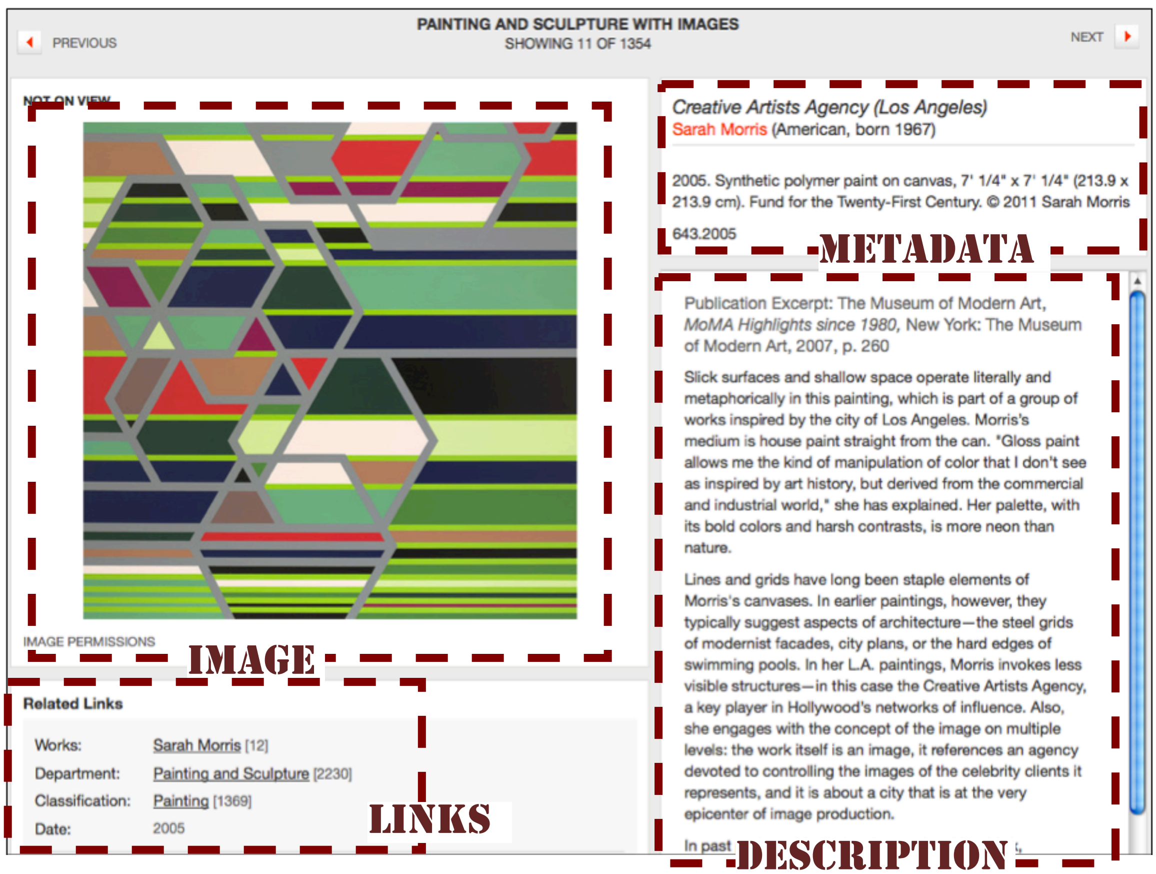

Fig. 1. Artwork detail from MoMA's Web page [MoMA 2011]

preparing for a test on modern and contemporary art on her home computer. She browses through a set of art works covered by her instructor in class that are hosted by the Museum of Modern Art’s (MOMA) web site. Figure 1 shows a snapshot of one of the artworks that Eva peruses online in preparation for her test. Each artwork is represented by its image, details of its composition (metadata), descriptive text that highlights the significance of the artwork, and a set of links for perusing related works. Interested only in doing well on the exam, Eva ignores the links section for now. As her alarm goes off, reminding her to leave for the test, she quits her Web browser and removes her home computer from her MIDAS device space. The MIDAS server, however, retains her browsing point.

As she walks out of the apartment, she connects to the MIDAS space through her smart phone and resumes studying. The bus is crowded, allowing her to use only the phone. The MIDAS system shows images on the phone’s display and converts the textual material to audio for playback, which she listens to, through a headset. She chooses to listen only to the description, skipping over the metadata and links sections.

Eva gets to her classroom with a few minutes to spare. With additional space and a desk, she opens her iPad and adds it to the MIDAS space. The system now uses both appliances to present the content, displaying the image and the description on the iPad and directing the metadata and browsing controls to her smart phone. Eva can now skim the text, while tapping on the phone’s display to move quickly through all the images before the test begins.

MIDAS’ ability to retain session data and co-use devices allows users to get the best renditions within the combined constraints of all their appliances. The rest of this dissertation is organized as follows: Chapter II surveys the prior work and relevant developments that inform this research. I outline my approach in Chapter III. Chapter IV presents 12C, a framework that I have created to describe and compare the properties of multi-device information systems. Chapter V introduces MIDAS architecture that my approach realizes. Chapter VI describes a formative evaluation related to human perception of image characteristics. Chapter VII presents the content selection policies for images that build on the results of this evaluation as well as policies for selecting media forms and for selecting target devices to present the selected instances. Chapter VIII analyzes MIDAS' properties with respect to the 12C framework, situating it in the context of other multi-device interfaces. Chapter IX concludes this dissertation and outlines trajectories for continuing this research.

CHAPTER II

RELATED WORK

The research presented in dissertation has been preceded and influenced by several research areas, among them hypertext, digital libraries, mobile and ubiquitous computing, and human-computer interaction. The following sections provide a brief survey of academic as well as industrial initiatives that have explored the issues involved in accessing information using multiple devices with diverse characteristics, simultaneously or not. I classify this related work into four categories: concurrent use of multiple devices, management of context for enabling users to switch appliances at will, adaptation of content for presentation within device capabilities, and the underlying hypertext infrastructure that these applications build on.

2.1 MULTI-DEVICE INFORMATION ACCESS

Various research projects have explored the possibility of supporting the co-use of information appliances both in the context of Web access and task-specific, stand-alone applications, for individuals as well as groups. Software that enables such co-use has been implemented as server-side enhancements, peer-to-peer client applications, or as middleware that mediates the flow of information between typical Web servers and typical Web browsers. Client enhancements have been explored mostly in the context of non-Web applications. Infrastructures for Web-based co-use have preferred Web proxies as the middleware grants significant autonomy to applications without requiring modifications to the Web server behavior.

WebSplitter was an early application to display XML-encoded Web pages on multiple devices, possibly belonging to different users [Han et al. 2000]. WebSplitter uses an XML framework that generates partial views of XML-encoded Web pages based on user permissions. The framework includes support for a proxy that enables users to split their view between multiple devices, such as desktops, notebooks, PDAs, and projectors. The proxy controls the channeling of the files associated with XML entities to a user's appliances; a static list of these is populated manually. The authors proposed to extend the WebSplitter architecture to include support for modifications to the device list as appliances become available or unavailable.

The multibrowsing framework enables users to move content across various computers including a user's personal clients, such as laptops, and PDAs as well as large screen displays in surrounding spaces [Johanson et al. 2001]. Multibrowsing supports various levels of commitment from clients. A user's devices may either participate as regular clients, enhanced clients or targets. Regular clients need no enhancements at all; they employ Web browsers without any extensions to direct specially encoded Web links to open on other devices. Enhanced clients install an application that enables them to redirect unenhanced Web pages to other computers. Finally, targets, such as public access large displays, must run an application that enables them to accept information that is directed toward them by regular or enhanced clients. While the multibrowsing infrastructure functions with client and server-side enhancements, the authors propose to employ proxies for improving portability of multibrowsing links, which are currently hardcoded.

The Ubiquitous Display System promotes the use of large, public access displays as extensions of mobile phones to display Web content that the phone is incapable of presenting [Aizawa et al. 2002]. The authors expect such displays to be placed in public spaces, such as "streets, malls, and shopping areas". A user in the proximity of a display may request a connection. Once connected, the user can open or close content on this display or interact with it by scrolling or following links visible on the display. The large screen is connected to a wired network and, in addition to enhancing the screen space, expands the bandwidth available to the user by downloading rich content directly from the Web via a wired network.

The projects described above channel content to devices that can present it best but do not adapt content to the device characteristics. This approach risks the possibility that no available device may be able to render some content, resulting in incomplete information availability to users.

The ANMoLE (Application-level Networking solutions for Mobile users) architecture delivers multimodal, multimedia content to device neighborhoods [Haneef and Ganz 2004]. ANMoLE envisions a variety of content adaptation mechanisms, such as conversion of text to speech, and the extraction of audio and visual streams from videos for routing each to a different device. This ambitious infrastructure spans wired, wireless, and phone networks to enable integrated content delivery to the appliance that is best suited for rendering each content element. The implemented demonstration prototype routed content simultaneously to a desktop, notebook computer, and a PDA.

The Interface Distribution Daemon (IDD) renders user interface elements over an interactive space of federated, network-capable, heterogeneous devices [Luyten and Coninx 2005]. The IDD is capable of splitting Web interfaces as well as generating cross-device stand-alone applications. The distributed Web interface is implemented as a Web server extension. The IDD supports multiple distribution modes: user-driven, where the user manually directs elements to particular devices; system-driven, where the user requests an interaction space and the system responds by transferring interface elements to detected devices; and continuous distribution, where the system redistributes elements each time devices join or leave the interaction space. IDD uses two metrics to determine the usability of an interaction space: completeness and continuity. In the IDD's task-centered model, satisfying the property of completeness requires that all interaction tasks required to reach a goal be accessible to a user using only the available devices. Achieving continuity requires that the user be able to interpret and evaluate the internal state of the system while using different interactive devices.

In the context of standalone applications, the Pebbles project explored the use of handheld computers in conjunction with desktop PCs. The project developed peer-to-peer solutions to support a variety of interactions between desktop and handheld computers including: extending the desktop interface to the handheld device [Myers et al. 2000b], using handheld computers as input devices for controlling applications running on the desktop computer [Myers et al. 2000a], and multiple PDAs as collaboration tool in conjunction with a desktop computer [Myers et al. 1998]. The

Pebbles architecture was developed with a primary focus on the Windows platform and it integrated mobile devices that ran Windows CE with Windows desktop computers. The project also developed an application to share data across platforms, such as Windows, PalmOS, and Unix, through the standard copy-paste command by synchronizing the clipboards of different computers [Miller and Myers 1999].

Other projects have explored the use of multi-device interfaces in areas such as computer supported collaborative work and rapid, multi-modal prototype development. Sánchez et al. [2010] propose the use of plastic interaction interfaces for a multi-platform, multi-device collaborative white board. They envision such a whiteboard to connect collaborators, who may be using diverse devices such as PDAs and high-end desktops, to work together effectively. The Interaction in Distributed Environments (INDiE) framework enables quick prototyping of interactive systems by coupling the capabilities of various input devices, such as PDAs and desktops, and sensors [Klompmaker et al. 2009].

The term "multi-device" has also been used to describe applications that are designed to run on multiple appliances but not simultaneously [Berti et al. 2004; Janousek et al. 2009; Meskens et al. 2010]. For example a NetFlix client that works on a variety of platforms and appliances enables users from any device with minimal re-learning because of the familiarity of its interface. While I acknowledge the benefits of applications that work identically (or similarly) on different platforms to minimize a user's cognitive load and to lower the barriers to switching devices, I use the term "multi-device" explicitly for referring to software that is designed for providing concurrent access from multiple computers.

2.2 SESSION MIGRATION

Dominant Web browsers, such as Firefox, Internet Explorer, and Safari, do not support portability of a user's browsing session. Users who have invested a significant amount of time browsing through a Web site or narrowing their choices during a search for something specific and now must switch devices are out of luck. They must restart their activity anew, supported by nothing but their recollection of the prior session. A robust body of research has explored mechanisms to retain browsing history as well as the current context of use when users switch appliances.

The browser session preservation and migration (BSPM) infrastructure enables users to take a snapshot of their active browsing state and retrieve it later on another device in order to continue browsing from this state [Song et al. 2002]. The BSPM is implemented as a browser-side plug-in that stores the snapshot to a proxy server. Another browser, also using the plug-in, can retrieve the snapshot to continue the user's session. The snapshot information includes the current page, browsing history, status of scripts, information entered in forms, and cookies.

The Internet Suspend/Resume (ISR) project at Carnegie Mellon University has developed software for enabling users to continue working seamlessly on their applications from different locations and devices [Kozuch and Satyanarayanan 2002]. ISR leverages distributed file systems coupled with virtual machines to save the state of applications and then resume these applications at a different time, different place, or from a different platform. The project uses the analogy of closing and opening laptops for suspending and resuming work without the necessity to carry the hardware for this functionality. Recent updates to the project have focused on using advances such as USB keys [Gilbert et al. 2010] and smart phones [Smaldone et al. 2009] as storage locations for saved information.

While the infrastructure provided by the ISR project enables users to continue their tasks, interruptions also result in increased cognitive load when resuming an activity. The Context Browser enhances user productivity by helping them visualize saved work contexts interactively via a timeline viewer and by retrieving a desired context to continue an earlier task [Park 2010]. Contextual information stored by the browser includes applications and processes running in a user's workspace, modifications to open files, and the locations open in Web browsers. The Context browser supports visualization of file activity, Web activity, and snapshots of desktop view. The browser also allows users to tag these views and enables searching for specific tags or for entire contexts by using nearness metrics to build a context by relating terms associated with applications that were in simultaneous use.

Cui et al. [2004] have developed a mobility-aware middleware to enable multimedia content delivery to users who switch between wireless (mobile) and wired networking contexts. This architecture stores application state in third-party location when accessing audio and video, thus providing support for existing multimedia content providers and users with few modifications to their infrastructure.

TERESA (Transformation Environment for inteRactivE Systems representAtions) is a software environment for designing "nomadic" applications—those that can migrate across devices and platforms [Mori et al. 2003], whether partially or entirely [Bandelloni and Paternò 2004]. The TERESA interface allows users to create XML-based top-down transformations to generate Web interfaces that render gracefully on appliances with diverse characteristics by applying specific filters to the abstract specification. The research group has continued this thread through projects that explore flexible migration of Web pages [Bandelloni and Paternò 2004] to enable Web users to migrate portions of Web pages to an appliance of their choice [Ghiani et al. 2010]. A visual interface supports Web users in migrating selected portions of the Web page to a particular appliance in order to continue the browsing session in mobile settings.

The MARIA language, an XML-based specification language, lets designers create model-based abstract Web applications that render differently depending upon appliance characteristics [Paternò et al. 2009]. The group has recently developed extensions to the MARIA language for specifying distributed user interfaces by migrating partial Web pages to different devices simultaneously [Manca and Paternò 2011]. This approach to designing applications that are geared toward migration is similar to that adopted by the User Interface Markup Language (UIML) [OASIS 2011] and by Grolaux et al. [2004], who distinguish applications intended for migration from those that are capable of it, for example, when applications running on Unix servers are opened remotely using X11 displays [X 2011] or the VNC remote control software [VNC 2011] these do not change their appearance based on the characteristics of the client device. These applications, while they are capable of migrating between clients with diverse characteristics are not designed to support such migration.

2.3 DEVICE-BASED CONTENT ADAPTATION

Applications that migrate between platforms or devices must also include mechanisms for adapting their content to various contexts. The issue is no longer restricted to the academic world, with platforms like Apple iOS, Google Android, and Microsoft Windows being used by millions of people on smart phones, tablet computers, and large screen TVs. Typical books on Web development for smart phones emphasize the design of device-aware applications [Frederick and Lal 2010]. Perhaps, the most obvious example of device-aware behavior is that of rendering large images and formatted documents on smart phones. On the flip side, presenting information that is designed for smart phones on larger displays is not straightforward either. In such cases the layout of the information elements as well as the elements themselves may be altered to forms that are better suited for the new medium.

Since its early days, the Web has been accessed from computers with diverse capabilities in terms of display real estate and network bandwidth. The Pythia Web proxy distilled images from Web page for presentation in low-bandwidth settings [Fox and Brewer 1996]. Distillation is a lossy, type-specific image conversion process that reduced the size as well as color-depth of the images in order to optimize the available network bandwidth. The system also enabled users to retrieve on demand the high-resolution image, or parts of it, a process the authors label as “refinement”. The Digestor system employed elision as well as transformation techniques to reformat Web pages dynamically based on device characteristics as well as user preferences [Bickmore and Schilit 1997]. Digestor applied a variety of techniques for manipulating images and text to adapt Web pages for presentation on a display size specified by the requesting client. These techniques include image scaling in pre-defined scaling factors (25%, 50%, and 75%), font size reduction, page outlining, and first sentence elision. While not as versatile as Digestor, Guirguis and Hassan’s [2010] framework adapts Web resources for mobile devices based on the characteristics of the individual device rather than the device class.

The Proteus Web personalizer goes a step further by adapting entire Web sites for mobile users to improve expected utility by analyzing the user's past behavior via server access logs [Anderson et al. 2001]. The power browser adapts textual content as well as forms for presentation on mobile devices [Buyukkokten et al. 2002]. Textual content can be displayed at different levels of granularity using expandable outline views or summarized. The power browser takes a novel approach to summarizing forms by displaying only the textual prompts associated with input fields while hiding the interactive widgets themselves. This allows users to scan the form quickly. The input widgets are displayed on request, when the user gestures over particular text prompts.

Multivalent documents are layered structures of related content: for example, image of a scanned page, positions of lines and characters, bibliography, structured content such as tables, images, and maps [Phelps and Wilensky 1996]. Users of these documents can add their own layers, called behaviors, to the documents. Behaviors interact with the content layers that they overlap, resulting in custom rendition of the document content. The multivalent browser, developed using this layered architecture, enables researchers to define custom behaviors for Web pages [Phelps and Wilensky 2001]. The browser extends the multivalent architecture to include support for HTML documents. It allows readers to superimpose behavioral layers to modify the standard Web page view. When layers cover parts of a Web page, only the content covered by the layer exhibits the behavior; for example, only a small part of the page may be magnified. When multiple layers overlap, the content exhibits the composite behavior of all layers that cover it; for example, if a translation layer partially occludes a magnification layer, the content in this area will be translated and magnified. This browser lowers the barriers to specify custom behaviors by allowing designers to create layers with simple behaviors and then to support complex behaviors based on interactions between these layers.

Another approach to device-based content adaptation is the generation of baseline representations that even the most resource-constrained devices can render. In the context of co-browsing, such a least-common-denominator baseline can be shared with all users to guarantee a degree of information availability for coordinating collaborative tasks. These shared viewpoints (SVPs) can be supplemented with personal viewpoints (PVPs) that contain additional detail or richer media for users who control more powerful computers [Chua et al. 2007].

Paternò and Santoro [2003] present a model-based approach to adapting Web content for desktops and data phones. This approach enables authors to decouple the design of information systems from their rendering on specific devices, albeit at a higher design cost. In a four-step process, a designer creates an abstract task model for an application, applies platform-specific properties to this model to generate a "system task model", includes user interface (UI) elements for accomplishing the various steps to complete the task, and then generates platform-specific representations of the abstract UI. This approach has been extended to include a semantic analysis of the interface elements as well as end user customization of the mobile adaptation specification [Paternò and Zichitella 2010].

Taking a conceptual view, Ma et al. [2000] propose a five-category classification of adaptation techniques: information abstraction (text summaries, image thumbnails), modality transformation (speech to text), data transcoding (color image to grayscale conversion), data prioritization (dropping low-priority content under resource constraints), and purpose classification (ads, banners, content, menus). Hoh et al. [2003] map an extensive array of techniques to this classification for text, image, audio, and video adaptation.

The World Wide Web Consortium (W3C) has promoted several initiatives to improve Web content adaptation for mobile devices. Most recently, the cascading style sheets 2.1 (CSS 2.1) specification was adopted as a recommendation [Bos et al. 2011]. Media types in CSS 2.1 enable content creators to describe context-dependent content and layouts, for example different content for desktop displays and mobile devices or removing background images and bright colors for laser printing. The client browser has the freedom and the responsibility to select the style sheet associated with the appropriate media type to generate the desired rendition. The next iteration of CSS, version 3.0, expands adaptation support to include media queries [Lie et al. 2011]. Acknowledging that mobile devices are not a monolithic class of devices, media queries enable authors to fine-tune their document presentation for devices that satisfy particular functional criteria. For example, authors may specify that displays must be at least 500 pixels wide or that the device include a color display or that a device possess a high-definition (HD) aspect ratio (16:9).

Before being discontinued in 2010, the Device Independent Authoring Language [Smith 2010] was the W3C's focus for authoring content that could be adapted for rendering in a variety of contexts. The language specification included an XML-based Web page structure, CSS modules, and form interactions. Extensions formalized in CSS3 overlap significantly in terms of the functionality that DIAL offered. Another W3C initiative, Composite Capabilities/Preference Profiles (CC/PP) [Klyne et al. 2010], lays out the W3C recommendation for expressing the properties and preferences of devices. CC/PP works in conjunction with User Agent Profiles (UAProf) [UAProf 2001] to describe device capabilities. Device Description Repositories [Smith and Sanders 2007] consist of such descriptions and expose public Application Programming Interfaces (APIs) that enable application developers to retrieve information about devices for content adaptation. The Wireless Universal Resource FiLe (WURFL), an open source, independent device description repository, contains information about thousands of devices and provides APIs in several major programming languages [WURFL 2011]. The repository includes information about the hardware, firmware, and software capabilities of thousands of devices. This XML repository employs a hierarchical structure that enables querying applications to get generic information about a device when specific information is unavailable. For example, when a phone manufacturer releases a new device that is not yet included in WURFL, the API may return information about infrastructure supported by the superset of the model, the manufacturer, or a generic mobile device. WURFL capitalizes on the high likelihood of manufacturers continuing their support for most-recently used operating systems or firmware in the newer models they bring to market. WURFL encourages community participation in updating the repository. Developers, manufacturers, and operating system makers are all encouraged to help fill gaps in the repository.

Synchronized Multimedia Integration Language (SMIL) is designed for interactive, multimedia Web presentations [Bulterman et al. 2008]. Authors can describe the spatial layout of visual objects, associate hyperlinks with media objects, and describe the temporal behavior of the presentation. In order to ensure the integrity of their multimedia presentations, SMIL enables authors to include alternate elements for rendering under specific conditions. Using the `switch` element, authors may include alternate media elements; each may be rendered under certain conditions. For example, a SMIL document may include high-resolution and low-resolution images or richer audio files for playing on clients using a broadband connection. When the presentation begins, the rendering client (browser) scans through the option list and selects the first media element whose constraints meet the system's characteristics. Subsequent matching elements are ignored.

2.4 HYPERTEXT ACCESS INFRASTRUCTURE

Device co-use requires information content transfer between multiple appliances, which is conducted using a variety of networking standards and protocols. While devices that communicate in peer-to-peer systems, such as Pebbles [Myers 2001], have the freedom to devise custom communication protocols, those that interface with the Web must adhere to Web protocols, such as the Hypertext Transfer Protocol [Fielding et al. 1999]. Many architectures use multiple protocols: for example ANMoLE [Haneef and Ganz 2004], WebSplitter [Han et al. 2000], and Ubiquitous Display System [Aizawa et al. 2002] must all support the HTTP protocol for requesting Web documents but must use a different, custom protocol for communicating between active devices. In the following subsections I describe the communication architecture used by the Web and a non-Web hypertext system, context-aware Trellis.

2.4.1 World Wide Web

HTTP, a generic, stateless communication protocol, underlies Web page transfer [Fielding et al. 1999]. Management of state allows a server to make decisions based on a record of prior information requests from a user (or a browser). State information could contain data about a user as well as her actions during the current browsing session as well as any historical browsing information. However, state management adds a significant overhead to server operation [Seebach 2008]. As a stateless protocol, the HTTP server trades functionality for efficiency. The inability of the bare bones Web servers to maintain state transfers this burden over to the Web clients and server add-ons. Cookies provide a mechanism for maintaining the state of a browsing session [Kristol and Montulli 2000]. Cookies are stored by Web browsers and are sent to the server as a part of the HTTP request. The server may use this information to retrieve data saved or operations performed during previous requests in order to provide the effect of a continuing browsing session. Thus, Web applications can take the advantage of session management when desired at the cost of added overhead.

The core HTTP interaction is based on request-response architecture. The client makes a request; the server responds to it and closes the connection [Fielding et al. 1999]. Support for persistent connections in the HTTP/1.1 protocol improves the efficiency of the server by avoiding repeated opening and closing of network connections but does not aid in the management of the state of browsing [Seebach 2008]. The request-response architecture does not allow the HTTP server to "push" information to the client(s) when updates relevant to these clients are available. In an increasingly mobile and connected world, users are often interested in social networking Web updates from their friends or entertainment or dining options in their vicinity. A Web application may alert mobile users when they are proximate to opportunities of interest or to the location of their friends. Web services can mimic a server push using a variety of sub-optimal techniques. The HTML standard supports short polling via the use of a `meta` tag, that instructs the browser to refresh the page at preset intervals [W3Schools 2011]. This approach, also called short polling, is viable but is dependent upon the browser requesting an update from the server. Consequently, if a state is updated just after the last response to a browser, the browser may have to wait several seconds to receive this update. HTTP supports long polling as well, where the server does not respond to a client request immediately but holds the connection open until updated information is available [Loreto et al. 2011]. An alternate approach, called HTTP streaming, keeps a connection open indefinitely. The server and client may send multiple requests and responses over this channel. While both these methods are used, RFC 6202 cautions developers about the issues involved. While taking no position on the use of these mechanisms, the document warns that their use may degrade server or network performance and provides best practices for prudent use. HTTP is clearly a protocol designed for performance and scalability.

Similarly, Web browsers are designed to gain wide acceptance. The browsers render a variety of media and have a large footprint in an effort to be everything to everyone. In order to ensure portability of user experience across platforms and browsers, these clients tend to conform to established W3C recommendations. This level of standardization is desirable for a system that is accessed by a large population but the hardcoded rendition of presented content raises the barrier for their adoption in research settings that explore new forms for expression.

2.4.2 context-aware Trellis

The Trellis project [Stotts and Furuta 1989] has explored the separation of aspects of hypermedia such as specification, content, presentation, behavior, and context of use [Furuta 2005] for investigating the structure and semantics of human-computer and human-human interaction in the context of hypermedia systems [Stotts and Furuta 1989] and computer-supported collaborative work (CSCW) [Furuta and Stotts 1994]. Trellis uses a formal, Petri net-based [Peterson 1981; Jensen 1992; Zurawski and Zhou 1994] representation of hypermedia documents and applications. In this section I have only tried to present the features that are essential for understanding my work and those that help punctuate the strengths of this system for design and use of document structures for multi-device browsing.

context-aware Trellis (caT) [Na 2001] extends Trellis by incorporating features that enable hypertexts to respond to characteristics of the users and their environment. For example caT may present different content or enable specific features for students and faculty; the system may direct users to an interactive chat during office hours or to a static help page in the evenings; local users may view different content from those accessing the documents remotely; or certain content may only be available after a critical mass of users is reached [Furuta and Na 2002]. While Web applications can (and do) accomplish such behaviors through specialized programming that is unrelated to the functioning of the basic HTTP server, caT's approach is remarkably different. caT enables these behaviors by leveraging the features of the underlying hypertext specification without the need for external programming.

caT hypertexts are represented as nodes and links—nodes represent locations that users browse to in order to view the associated information content and links represent connections between nodes that a user can browse along to reach other nodes, much like HTML links. The node-link structure that forms the structure of caT hypertexts is clearly distinguished from the content that is displayed when particular nodes are reached. The information to be displayed is stored in its own file and is associated with nodes in the hypertext structure. This relationship is similar to the `img` tag in Web documents, which binds an image stored in an external file to an HTML page [Hickson 2011; Raggett et al. 1999]. In contrast, the paragraph or `p` tag encloses the content to be displayed. Thus, Web document authors can replace images associated with a document by changing the image file while retaining its name but must modify the HTML document to edit the content of the paragraph text. caT's separation of content and structure is similar to that offered by Web technologies such as XML Stylesheet Language Transformations [Clark 1999]. caT authors may modify the structure of the hypertext, thus affecting the links that may be available to a reader from various locations, without ever changing the information that is displayed for any of the places.

Architecturally, caT (and Trellis) is based on a client-server model, much like the Web. However, unlike HTTP servers, the caT server is stateful; it maintains the state of browsing for all users connected to it, allowing browsers to join and leave as they please without resulting in a loss of the browsing state. Indeed, the state is maintained even when no browsers are connected to the server. A user may connect to the server and pick up browsing as if the current browser was the one that initiated the session. A user may also browse simultaneously from multiple browsers that run on multiple computers. The state of browsing is accurately reflected in all browsers. Change to the browsing state, whether caused by a user's actions in one browser (such as, following a link) or due to changes in a user's environment are propagated to all browsers in order to maintain consistent views on all connected browsers.

caT enables, and indeed promotes, the development of special-purpose browsers that explore a variety of mechanisms and modalities for data presentation. These include xtb2 [Na 2001], a text-centric browser that allows document authors to specify helper applications for rendering non-textual content, a Web browser, an audio browser [Ustun 2003], and a spatial browser [Karadkar et al. 2004], which renders information on a canvas while maintaining spatial relationships between content elements. Unlike Web browsers that render content received from the server homogenously, caT browsers have significant presentation autonomy. The textual browser ignores the spatial elements of content that it receives, displaying only the text. The audio browser converts this text to an audio stream. Thus, while the hypertext specification separates structure of a document from the content associated with it, browser autonomy separates the content from its presentation.

xTed, an authoring tool as well as a browser, is capable of rendering a caT hypertext's underlying net structure. The other browsers described above present only the contents associated with the hypertext. Hypertexts that reflect real-world constraints or those that present large-scale information tend to grow rapidly and quickly become too unwieldy to be authored using xTed's bottom-up approach of node-and-link authoring. TcAT (Template-based caT Authoring Tool), provides a mechanism for top-down, function-oriented authoring by allowing authors to specify higher-level structures and behaviors, for example authors may specify that certain elements be displayed simultaneously (in parallel) or serially [Park et al. 2010]. TcaT also identifies repetitive sub-structures and aids the authors in optimizing their hypertext structures for improved performance and ease of maintenance.

This dissertation extends the expressive capabilities of caT further, by introducing multi-browser, coordinated, device-based content adaptation.

CHAPTER III

APPROACH

Many projects that have developed distributed user interfaces stress the need for dynamic reconfiguration when current devices become unavailable or new ones become available [Han et al. 2000; Johanson et al. 2001; Haneef and Ganz 2004]. The Interface Distribution Daemon describes design details of an automatically reconfiguring multi-device display system [Luyten and Coninx 2005].

3.1 THE 12 C'S

Reconfiguration of distributed interfaces involves several issues. I begin with an articulation of these issues and organize them in a '12C' framework. The 12 C's are:

- Concurrency – presenting information simultaneously on different devices
- Control – centralized or distributed rendering of actionable elements (links)
- Comity – homogeneity in device properties
- Completeness – presenting all required information elements
- Coverage – routing information elements to devices for optimal presentation
- Conversion – changing the form of information content for optimal coverage
- Composition – mechanisms for creating a coherent presentation
- Coherence – ensuring that information presented is consistent
- Coordination – redistributing information elements to devices when necessary
- Continuity – ensuring that users can continue the tasks that are in progress when the presented information changes
- Constancy – presenting information in forms that have been used before to leverage human memory
- Confidence – aspects of trust and security in cross-device communication

These issues address critical aspects for designing features of multi-device systems, such as infrastructure, policies, interface, user experience, and security. The twelve dimensions act as an exploration space for describing, comparing, and designing multi-device systems. The framework is a descriptive tool only and systems may be designed that address only some of these dimensions.

3.2 IMPLEMENTATION PLATFORM

MIDAS extends caT. While the Web presents a compelling case as the dominant, globally accessible and accepted hypertext system, its architecture limits true bi-directional communication, which is necessary for real-time coordination of information presented on different devices. In contrast, caT provides a reliable mechanism for pushing information to clients.

caT's simplicity in separating structure from content also weighs in its favor. MIDAS leverages the association of content with structure by introducing a level of indirection that enables it to customize the content that is delivered to devices based on their characteristics. Thus, depending upon device characteristics, MIDAS may present one with an image and present another with a text file (presumably, one that is conceptually related to the image). Pointers on the Web—such as links, images, or SMIL targets—refer to individual files and replacing one file with another of a different type is inconvenient, at best.

As the designers of the multivalent browsers expressed, Web browsers do not lend themselves to designing unusual behavior or communication that is necessary in a research setting [Phelps and Wilensky 2001]. In a similar vein, a point that weighs in caT's favor is the ability to design quickly small, lightweight browsers or interfaces with custom behaviors. I demonstrate this ability by implementing a Web interface for MIDAS. This interface enables Web users to interact with MIDAS via a typical Web browser, such as Firefox or Internet Explorer. This experience remains constrained by the inability of a Web browser to push information back to the browser, an issue already discussed above. While not a perfect solution, this browser forms a first bridge between the popular Web and the expressive caT.

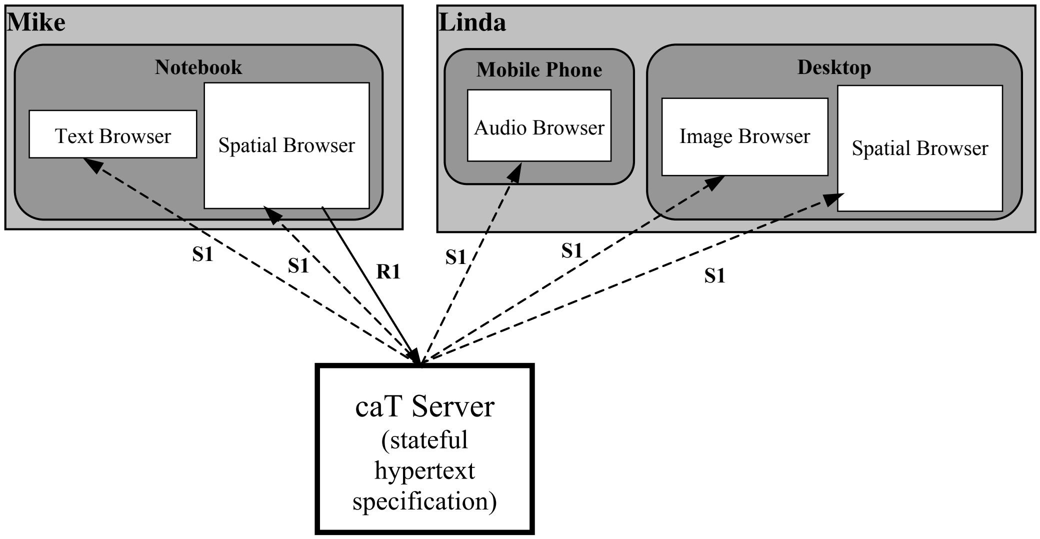

Fig. 2. caT communication

3.3 ARCHITECTURE

The MIDAS Web interface—a convenient and lightweight tool for bringing MIDAS services to the Web—is not the only avenue I envision for taking advantage of the Web's global reach. While MIDAS extends caT, it is implemented as middleware that positions itself between the server and the clients, much like a Web proxy. Figure 2 illustrates the communication between a caT server and the clients (browsers) connected to it. Mike and Linda are co-browsing a technical document that they are reviewing. Mike is working on his laptop, while Linda is using her desktop. They both use caT's spatial browser to view a complex schematic diagram that links together various aspects of an engineering system in multiple, inter-connected layers. To maximize the space available for viewing the schematic diagram, Linda connects to the caT server from her Linux-powered smart phone and starts up an audio browser, which plays her the textual content in the document. When Mike follows a link included in the schematic, the request R1 is sent to the caT server, which responds by sending the updated contents to all the connected browsers. The message S1 is sent to both of Mike's as well as all three

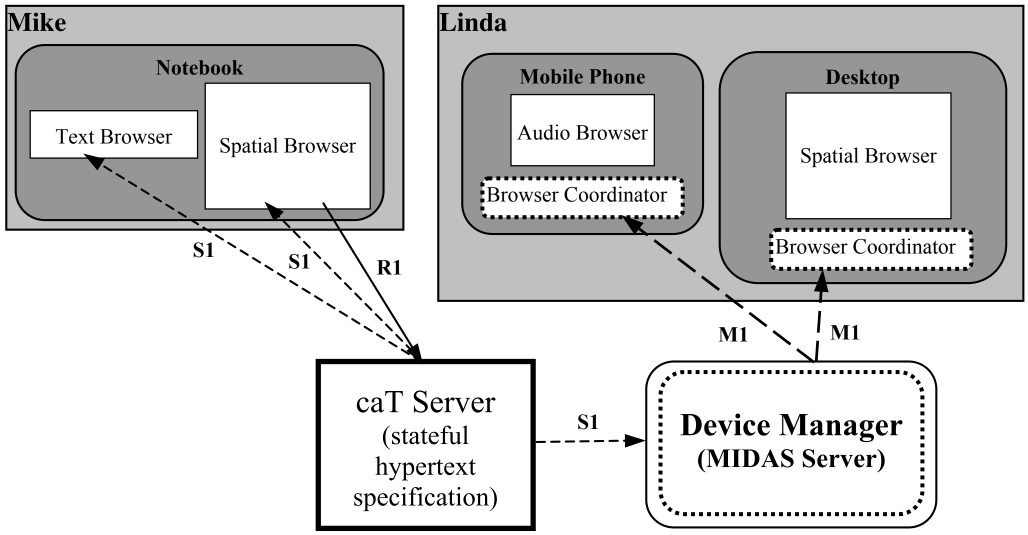

Fig. 3. caT-MIDAS communication

of Linda's browsers. As there are no images associated with the new state, Linda's image browser does not display any content.

Figure 3 illustrates how this scenario changes with the introduction of MIDAS. Linda, who is using multiple computers, adds them to the MIDAS space by starting a program called Browser Coordinator on her desktop as well as her mobile phone. The Browser Coordinators communicate with the Device Manager. This MIDAS infrastructure now negotiates the communication between Linda's devices and the caT server. As far as the caT server is concerned, the Device Manager is just another caT browser connecting to it (notice that the response S1 is now sent to the Device Manager). The Device Manager determines the most appropriate content available for Linda and routes it to each device. It acts as the centralized presentation coordinator that resolves conflicts and ensures that critical information is presented. Linda does not have to start or quit individual browsers. Since there are no images to display, the Browser Coordinator does not start the image browser on Linda's desktop. Just as the caT server considers the Device Manager as a typical caT browser, the caT browsers communicate with the Browser Coordinator as if it were a caT server. MIDAS components thus neatly insert themselves within caT's client-server communication.

This architecture provides for the greatest flexibility in working with different server and client systems including the Web. For example, the MIDAS Web interface could just as well interact with a caT server. Similarly, the design of the Device Manager could easily be extended to communicate with a Web server by changing the communication protocol but without having to change the data representations or policies.

3.4 ENVIRONMENTAL INFRASTRUCTURE

MIDAS enables users to present information on all computing devices available to them. These may include networked personal appliances carried by a user or, when available, computers or displays in the user's environment. While MIDAS can integrate devices in public places when these are available, the architecture neither requires nor expects that a user's surroundings have specific computational or sensory enhancements. Services that rely on specific enhancements to a user's surrounding limit the areas in which they can be effectively used. To minimize potential points of failure, MIDAS limits its reliance on external infrastructure.

A user must intentionally integrate a computer in her MIDAS space, whether it is her personal appliance or one available to her temporarily. Thus, a user's appliances are not automatically available for MIDAS' information presentation. She may choose to share the displays of her notebook and tablet computers, leaving her smart phone unconnected to MIDAS when she is expecting important calls so can answer them without compromising the integrity of her MIDAS presentation.

CHAPTER IV

12C: A FRAMEWORK FOR MULTI-DEVICE USER INTERFACES

Systems that distribute their user interface across multiple devices are complex by virtue of the diversity of their properties related to infrastructure, policies, and users. Several projects, reviewed in sections 2.1 and 2.2 have focused on designing and evaluating distributed user interfaces (DUIs). While these explorations are valuable for establishing proof-of-concept systems that afford specific features, the characteristics of these systems vary significantly from each other. When a variety of systems that support distinct features and adopt differing approaches populate a conceptual area, models and frameworks help mark the boundaries between these system by serving as tools for description, classification, comparison, analysis, and the design of future systems.

Such models and frameworks have been designed for systems in areas such as hypertext systems, digital libraries (DL), computer-supported cooperative work (CSCW), and tangible user interfaces (TUI) as each of these areas evolved. The Dexter hypertext reference model captures important abstractions found in hypertext systems implementations with a view to providing a methodical basis for comparison of these systems and to enable interchange and interoperability standards [Halasz and Schwartz 1994]. In the domain of digital libraries, the 5S framework provides a formal theoretical model and serves as an aid for designers, implementers, and evaluators [Gonçalves et al. 2004]. As the field of research progresses, models evolve too. CSCW systems have long been classified along the space-time array, which yields four possibilities based on synchronous or asynchronous application of each dimension [Johansen 1988]. This simple, yet (or perhaps, hence) effective model has been extended by Penichet et al. [2007] to address the increasing complexity of groupware systems by including additional characteristics, such as information sharing, coordination, and communication among users. Similarly, Ullmer and Ishii [2000], proposed a conceptual framework for TUI systems. They classified TUI systems based on a set of characteristics. As this initial classification did not include humans interacting with intelligent systems, van den

Hoven and Eggen [2004] have proposed extensions that include behaviors and characteristics embodied by their system.

Luyten and Coninx [2005] have proposed completeness of interface elements included in a presentation and continuity of user interface as the two candidate metrics for determining the usability of distributed interaction spaces. I posit that a critical point that requires taking a high-level survey of the issues involved in DUI systems has been reached. I present the 12C framework that articulates issues pertaining to DUI infrastructures, policies, and usability issues. This framework provides a template for describing the characteristics of and comparing DUI systems.

4.1 CONCURRENCY

Concurrency refers to the property of a system to present information simultaneously on multiple devices. While infrastructures such as ANMoLE and the IDD enable synchronized multi-device presentations, the early versions of TERESA and the Web session migration initiatives do not. Systems may support concurrency with some restrictions: inclusion of public devices is one such scenario. While ANMoLE does not include public devices, the Ubiquitous Display System enables the incorporation of public devices. Automatic versus interactive inclusion of devices is another aspect of concurrent presentation. Systems such as IDD, WebSplitter, and ANMoLE automatically include all the devices that are available to a user and select those to use for specific tasks. As an example of interactive inclusion, the Ubiquitous Display System requires a user to interact with the system in order to include a public device in her space. A final aspect of concurrent behavior is whether this behavior requires interaction. In the case of the Ubiquitous Display System, a user directs content that cannot be displayed effectively on a mobile phone toward the public display. The multibrowsing framework includes specially encoded links and locally installed software, either of which channel the target documents toward other devices present in the space. In contrast, ANMoLE and WebSplitter both generate multi-device displays automatically, by mapping the characteristics of the documents to be presented to those of the available information devices. The IDD supports both user-driven and system-driven approaches to interface element distribution.

4.2 CONTROL

Just as an infrastructure may present documents on multiple appliances, it could also distribute controls—elements for interacting with the presentation, such as links, forms, and menus. Systems such as WebSplitter distribute the documents as well as the controls, while the Ubiquitous Display System, which employs public displays to augment the visual properties of a user's mobile phone, uses a centralized model. The controls reside entirely on the mobile phone. While these two demonstrate the extremes: complete centralization vs. complete decentralization, an infrastructure could restrict the distribution of control elements to devices using certain platforms or those that possess certain display properties or particular interaction modalities. For example, a system could restrict the presentation of forms to devices equipped with full keyboards. The infrastructure could also remain agnostic on the issue and allow individual devices to determine whether each would be a control device.

4.3 COMITY

Some systems are designed to work only on certain platforms, for example Pebbles works on devices that run Windows. Recent gaming applications, such as multi-device Rummy and Poker are designed for Apple's iOS [Döring et al. 2010; Green 2010]. In contrast, ANMoLE is designed to include information devices that work on different platforms, consist of various form factors, and also includes appliances such as television screens and music systems. The multibrowsing platform is capable of incorporating diverse devices as enhanced clients and targets when a local application to support the desired interaction is installed; inclusion of other devices as regular clients needs no enhancements to the device at all.

Multi-device systems may be designed to accommodate devices with certain physical characteristics or those that support particular protocols. For example an infrastructure may integrate tablet computers appliances with multi-touch screens. Another may require that target devices support Bluetooth or Wi-Fi.

In general, the reach of a system in terms of potential devices is inversely proportional to its expectation of comity with respect to specific attributes. Clearly, not all attributes are equally restrictive. Most networked appliances, including modern smart phones, support Wi-Fi as well as location-awareness add-ons, such as global positioning system (GPS) sensors but few support new platforms like Windows Phone [Microsoft 2011b].

4.4 COMPLETENESS

Web pages include specific content instantiations to be presented, whether as text, images, audio or video files as well as the spatial relationships between these elements by virtue of the HTML specification. Pages are delivered as a single unit; the author controls the content that is delivered to viewers. When information is rendered on multiple devices, there are few conventions that authors or systems designers can adopt. Systems as well as authors must include flexible mechanisms for rendering information consistently in the face of a changing set of devices. There is, thus, potential for exploring a range of behaviors.

The approach adopted by Chua et al. [2007] generates minimal representations that can be rendered on any device and supplements these with additional detail for presentation on laptop and desktop computers. This unique, bi-level approach for completeness guarantees that each user will receive all the information elements in some form. Ghiani et al. [2010] demonstrate a system for enabling viewers to generate partial views by selecting portions of logical descriptions of Web pages. This approach engages the viewer in the process of defining completeness alongside the author, albeit within the initial specification generated by the author. Similarly, other specifications could be adopted where the form may not change but the content or the hypertext structure itself is associated with identifiers that allow the author to indicate whether the content element is necessary to present a semantically meaningful view to the reader or is optional.

Luyten and Coninx [2005] argue that it is impossible to ensure completeness in dynamic distributed system. This argument remains valid when a system relies on fixed content representations and interaction modalities for information elements. However, allowing for flexibility in rendition of content enables a system to guarantee completeness, if only by compromising on the quality of individual content elements.

4.5 COVERAGE

This attribute describes how well a system presents information within the constraints imposed by the available devices. Providing good coverage ensures optimal use of available resources such as display real estate, network bandwidth, disk space, and processing power. In systems such as WebSplitter and IDD with system-driven distribution that use automated algorithms for enabling concurrency this metric expresses which device properties the system optimizes the coverage for. Systems may develop quantitative metrics for comparing the relative merits of various options for presentation of content elements. Determining the quality of coverage could be a tricky venture as optimizing coverage for a document could result in degradation of coverage for others. Thus, a system may have to balance the metrics for coverage of individual documents against that for an optimal global view of all presented information. Another potential hazard is that a desktop computer may be the best possible option for displaying all information elements and other “lesser” devices may starve.

Infrastructures that enable users to transfer contents across devices manually could also employ a coverage metric. In such cases, the metric could be used to inform users passively about the efficiency of the current presentation and potentially, alternatives for improving the coverage.

4.6 CONVERSION

It is also possible that none of the devices available to a user may be able to render a document. In such cases, the infrastructure may include features for converting these documents in order to present them within the constraints imposed by the available devices. Such conversion mechanisms may modify the content while retaining the form, for example text summarization or may retain the content but change the form, for example, text-to-speech conversion. Scaling of images and extracting text from PDF documents are other popular methods for content conversion. Hoh et al. [2003] map an extensive array of techniques for content conversion and to Ma et al.'s [2000] classification of adaptation techniques.

Conversion may be performed while authoring content or at when documents are requested by the viewers. Documents can be converted at presentation time with the knowledge of the context in which they will be rendered: device characteristics, user preferences, and environmental properties. However, transforming documents on the fly may delay response time, potentially leading to a less satisfactory user experience. On the other hand, documents could be converted when they are added to the system for ready delivery in a variety of contexts. While this approach enables quick turnaround time in document delivery it suffers from the drawback that it is rarely possible to foresee all the possible contexts in which documents may be requested. Thus, it may restrict users to unsatisfactory renditions in contexts that were not envisioned by the infrastructure designers. These approaches can be merged by converting documents when an appropriate option is unavailable and storing all converted documents for future use.

Ensuring that the converted documents maintain the integrity of information is a critical task also. While necessary, ensuring that the converted forms adhere to the "spirit" of the source document is a resource-intensive and time consuming process as it often requires a human being to judge the appropriateness of the resultant documents. Infrastructure developers and document authors may develop procedures for personally verifying a subset of the generated forms to evaluate their suitability for presentation in lieu of the source document.

4.7 COMPOSITION

The composition of multi-device presentations may take a degenerative or generative approach. In the degenerative approach, content authored for rendition on a single device is split and presented on multiple devices. For example a pre-authored Web page may be partitioned into several components, such as the navigation menus, images, textual content, links, and advertisements. Some of these components may then be displayed on a mobile phone and others a tablet computer; possibly, one or two may be presented on both devices. Two major approaches for disintegrating Web pages are DOM-based content extraction [Gupta et al. 2003] and visual page segmentation [Cai et al. 2003]. DOM-based approaches rely upon the structure of HTML documents to infer relationships between page elements. Visual segmentation mechanisms partition pages based on a spatial layout of the page, attempting to mimic the human visual sense-making processes.

A variation of this approach is a filtering-based approach, where portions of a document may be filtered either manually or algorithmically to generate partial views for a variety of devices. Using algorithms that match device characteristics to those of the content and users, WebSplitter filters XML-based Web page specifications to generate device-specific renditions [Han et al. 2000]. Ghiani et al.'s system [2010], supports manual filtering of content for migrate portions of Web pages to mobile devices.

A generative approach, on the other hand, combines separately authored components for presentation on multiple devices. In this approach, a "master" document that presents a precise global view to be presented during a browsing session may not exist prior to the rendition. What a generative approach lacks in specificity, it makes up for in flexibility. While rule-based filtering or splitting approaches may need to be tweaked when new devices, presumably those with unusual characteristics, are used to access the information, documents generated without a pre-defined ideal have greater flexibility in adapting the presentations for unforeseen situations.

4.8 COHERENCE

Document content, presentation forms, data formats and interaction modalities available to the user must all mesh together to create a well-designed rendition. Crafting a well-designed presentation in a multi-device context with an unpredictable set of devices is a significant challenge. This attribute describes the homogeneity and consistency of the rendered information. Coherence is a subjective metric that requires human evaluation. Multi-device presentations created using a degenerative process may use the initial document as a baseline for evaluating the quality of the composed rendition. However, the multi-device presentation may not be directly comparable to the source due to changes in the properties of presentation devices and interaction modalities. Generative presentations, on the other hand, could not be compared against a pre-set standard and authors must judge the impact of these presentations independently.

4.9 COORDINATION

Content presented on the participating devices must continually be coordinated when user actions result in changing the state of browsing. The information content and interactive elements associated with the new state must be propagated to the devices. In systems that include user preferences or real-world constraints such as location and time, the state of browsing may change without any action on part of the user. For example a user's option to initiate a live chat session with a helpdesk employee may be rendered unviable when the workday ends. If the user had opened the document during the workday, this link must now be removed or disabled. While taking away this option, the system may also highlight online, self-guided help options, resulting in a change in information content.

In additional to user initiated and environmental causes, multi-device presentations may also require coordination when the set of available devices changes. A user may wish to include additional devices or active devices may cease to function. Regardless of user intention or choice, the system must attempt to conserve the coherence of the presentation by reallocating information elements. The availability of a new appliance provides additional options to the coordinating software for presenting information elements. In this case, the device does not threaten the integrity of the information content that is currently presented to the user. The coordinating agent may safely defer content reallocation until the next change of browsing state. However, this lack of action would result in the display real estate as well as the interaction modalities of the new device remaining unused till it is included in the presentation.

When a device ceases to participate in a browsing session, the situation may be critical. The contents as well as interactive elements rendered by this device become inaccessible, resulting in a loss of completeness and likely resulting in a loss of coherence in the presentation. In this case, the coordinating agent must remedy the situation by rerouting the lost information elements to other devices.

Alternately, the coordinating agent may reallocate all information elements associated with the current browsing state whenever a device leaves or joins the presentation, resulting in an optimal use of the available display real estate and interaction modalities at all times. However, repeated reconfiguration of elements in a browsing session may cause disruption in the user's task as she must now get her bearings in the reconfigured presentation.

4.10 CONTINUITY

Continuity of the user interface, another metric proposed by Luyten and Coninx [2005], emphasizes presentation of content in a manner that enables users to interpret and evaluate the internal state of the system. Multi-device systems may support users in this goal by reallocating only the necessary elements and only when necessary. Thus, in order to improve continuity, the coordinating agent may adopt the approach of migrating the contents rendered on a device that is no longer available without affecting the information elements already presented by other devices. This approach sacrifices optimal use of resources in favor of reducing the user's cognitive load. Furthermore, systems may be designed to maintain the form and location of information elements that remain stable when the browsing state changes. Thus, elements that are displayed would undergo minimal modifications. However, striving to maximize continuity is likely to affect the coverage of a browsing state as the optimal views may be ruled out by elements that occupy prime space on a target device.

4.11 CONSTANCY

While continuity focuses on aiding users in assessing the state of a system, constancy leverages long-term memory to present information in forms that it was previously rendered in, whether during the current browsing session or in the past. When a user revisits a hypertext, she may see images or text that she has seen before, within the constraints imposed by the available devices. Keeping track of the forms that users have seen may add a monotonously increasing overhead with time. Also, when available devices support the presentation of only some documents that a user had seen earlier, preferring these to the most optimal forms for the current devices may negatively impact the coverage of these elements.

4.12 CONFIDENCE

To maintain the confidence of users in the information being presented, the multi-device infrastructure must maintain secure communication channels with trustworthy devices. The system could include mechanisms for authenticating users as well as devices. While security issues in maintaining reliable communication channels are numerous, they are outside the scope of a thorough discussion here.

4.13 12C COMPARISON OF MULTI-DEVICE SYSTEMS

Table I compares the characteristics of multi-device systems reviewed in Chapter II according to the 12C framework. With the exception of caT, my assessment of the properties of these systems is based on the published reports about these systems and not on experience with using the system itself. Systems have explored the support for aspects of concurrent displays as well as distributed and centralized control. When the published work is either ambiguous or silent on an aspect of the system, the table indicates the lack of information with a question mark rather than guessing the system designers' intent.

Few systems are designed for appliances with specific characteristics. While many infrastructures rely on client-side software for participation in device neighborhoods, this is not an inherent restriction based on device characteristics. As a practical constraint, designers may create client software for certain platforms in order to evaluate the performance of their systems. In contrast, the Ubiquitous Display System is designed for including large public displays in a user's personal device space.

The "Coverage" property does not apply to systems that require manual transfer of contents between devices. While most systems do not support content conversion, those that rely on manual inter-device content transfer can guarantee completeness of the presented content.

Similarly, systems that rely on manual content transfer support the continuity of user tasks by involving users in the process of content migration. Systems that do not convert content forms also support constancy robustly as the content to be presented to users has been preprogrammed either in the specification or in the policies.

The next two chapters discuss the design of the MIDAS architecture and policies. Following this description, I will discuss the characteristics of MIDAS in the context of the 12C framework and compare the design decisions in the context of other multi-device systems.

Table I. Characterization of multi-device systems in the 12C framework

12C Attribute	WebSplitter	Multibrowsing	Ubiquitous Display System	ANMoLE	IDD	Pebbles	context-aware Trellis
Concurrency	automatic personal	manual public	manual public (2 devices)	automatic personal	manual, automatic personal	manual personal	manual personal
Control	distributed	distributed	centralized	?	central, distributed (strategy-dependent)	application-dependent	distributed
Comity	unrestricted	unrestricted	large displays	unrestricted	unrestricted	Windows	unrestricted
Completeness	?	Y	Y	N	N	Y	browser-dependent
Coverage	Y	N/A	N/A	Y	Y	N/A	N
Conversion	N	N	N	Y	N	N	N
Composition	filtered	generative	generative	generative	filtered	N/A	generative
Coherence	Y	Y	Y	Y	?	Y	browser-dependent
Coordination	User	N/A	N/A	User	User, Device	User	User, Environment
Continuity	?	Y	Y	?	Y	Y	Y
Constancy	Y	Y	Y	N	Y	Y	Y
Confidence	N	N	Y	Y	N	N	N

CHAPTER V

MIDAS SYSTEM ARCHITECTURE

Multi-device Integrated Dynamic Activity Spaces (MIDAS) is a middleware extension for client-server information systems. As its name indicates, MIDAS creates an interactive space by integrating presentation and interactive modalities of all devices available to a user for unified content delivery to these devices. This approach is similar to that adopted by the virtual spaces in IDD [Luyten and Coninx 2005] and the integration of input modalities in INDiE [Klompmaker et al. 2009]. By integrating the interaction layer of many devices, MIDAS can deliver information in forms best suited to the available modalities regardless of the physical devices that host each mode.

5.1 OVERVIEW

MIDAS interposes between the client(s) and the server of an information system. This approach enables MIDAS to interact with different information systems by developing the appropriate stubs for communication with the clients and the server without modifying MIDAS' internal representations, which are abstracted into two categories: interactive (links, form inputs) and non-interactive (images, PDF files, text). In addition to the approach, MIDAS' implementation is also geared for multi-platform use. Programmatic components are implemented in Java and use TCP socket communication. Textual configuration files and a MySQL database backend for structured data round out the platform-neutral implementation.

As shown in Figure 4, the MIDAS architecture comprises of three components: Device Manager, Browser Coordinator, and Resource Realizer. The Browser Coordinator is the "app" that a device must run in order to be included in the MIDAS space. The Browser Coordinator maintains a registry of the device properties as well as the media formats that it supports and other software, such as browsers, that it can employ for presenting content. The Device Manager coordinates the communication between the information service (caT or a Web server) and the Browser Coordinators.

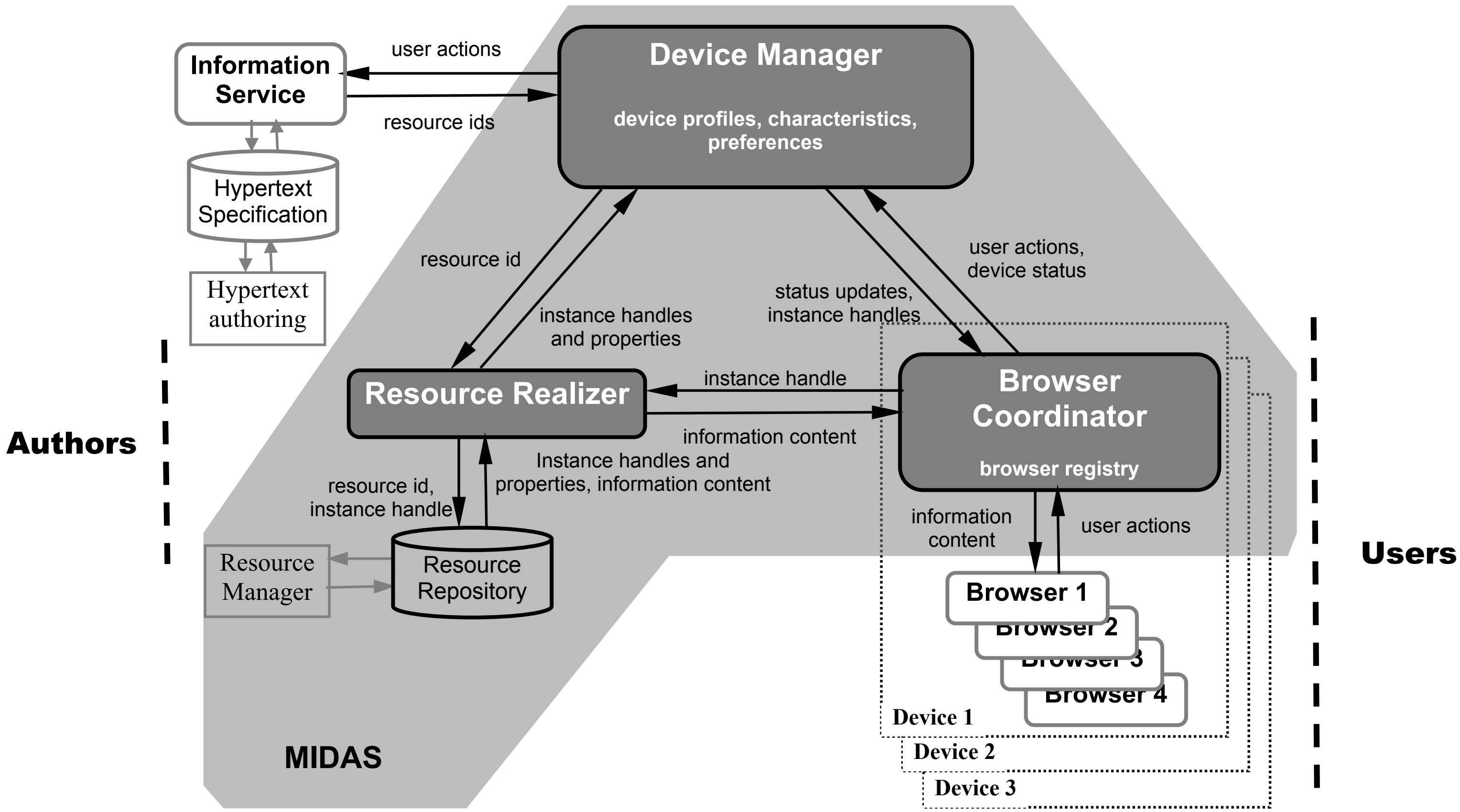

Fig. 4. MIDAS system architecture

Functioning as the brain of the MIDAS architecture, the Device Manager routes information to the available devices. MIDAS maintains a registry of device capabilities and retrieves the properties of content in its various forms from the Resource Realizer. It then maps the properties of the content to those of the devices and selects the information elements to be presented on each device. The Resource Realizer maintains a repository of the content elements in various forms and serves content to the Device Manager as well as the Browser Coordinators upon request.

5.2 INFORMATION SERVICE (caT)

The Information Service is a hypertext engine. It reads the hypertext specification, effects user actions received from the browsers on this specification, and returns the resulting state back to the browsers. Currently, the MIDAS infrastructure includes a socket interface for communication with caT servers and clients, owing largely to caT's support for server-side state management and a reliable push mechanism. caT's separation of content from structure is yet another point in its favor. Attaching documents to specific locations within the hypertext structure can be easily extended to add another layer of indirection—an abstract "resource" that serves as a placeholder for content instantiation. A unique resource identifier (id) serves as a handle for locating the information associated with any resource. MIDAS exploits this flexibility to enact display agnosticism [Karadkar et al. 2004] in order to deliver content based on characteristics of the target device.

The resource id groups various instantiations of information contents that may be used interchangeably. The resource id thus acts as an abstract representation of equivalent information content representations and encapsulates them into a semantic unit. caT hypertexts refer to information content via these resource ids. The Device Manager resolves this abstract identifier to a specific expression of the content element. In programming language parlance, the resource handles serve to delay the binding [Gantenbein and Jones 1986] of information content with the hypertext specification.

MIDAS thus introduces polymorphism [Eckel 1993] by allowing the Device Manager to bind the information content with the hypertext at display time.

As far as the caT server is concerned, the Device Manager is just another client that it serves. The fact that the Device Manager requests the resource ids and not content handles—as other caT browsers do—is a testament to the autonomy that caT bestows upon its clients. Indeed, MIDAS works in conjunction with other caT browsers in order to affect and be affected by users' actions in these browsers. Thus, the MIDAS infrastructure also receives updates from caT that are caused by environmental factors, such as change in time or location.

5.3 DEVICE MANAGER

The Device Manager, together with Browser Coordinators that run on MIDAS client devices form the information routing mechanism. As shown in Figures 4 and 5, the Device Manager communicates with every other MIDAS component and is the heart of the system. It receives the current state of the browsing from the caT server and stores the resource ids for the active presentation elements (contents) as well as interactive

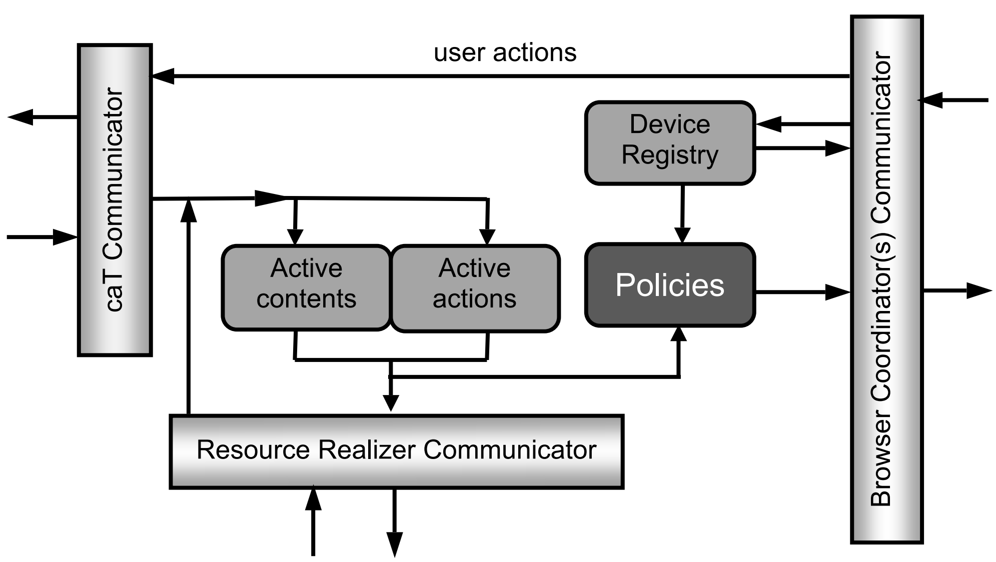

Fig. 5. Device Manager architecture

elements (actions). It communicates with the Resource Realizer to retrieve the properties and handles for all available instantiations. In addition to the set of active information elements, the Device Manager maintains a registry of active devices. The Browser Coordinator running on each device registers itself at start up and unregisters when it shuts down. The device Manager also serves as a conduit for the communication between the caT server and browsers. It forwards user actions, such as following links, to the caT server. It also returns the information to be presented to each device, although that is a more involved process.

The Device Manager implements a set of policies (discussed in detail in Chapter VI) to map the resource instances to characteristics of the rendering devices. Using the information stored in the device registry, it maps contents to devices and returns to each device the target resource handles for content and actions. In effect, the Device Manager acts as the centralized presentation manager for the MIDAS architecture.

5.4 RESOURCE REALIZER

The Resource Realizer manages the multiple instantiations of information elements and acts as an interface that enables other MIDAS components to access these resources. A resource id encapsulates resources that contain similar or interchangeable information. For example, photographs of Michelangelo's David, a video related to the creation and upkeep of David's statue, and a textual transcription of the video narrative as well as some published works about this masterpiece may all be stored as a single resource or conceptual unit. The Resource Realizer connects the resource ids to the various files that contain the content associated with this id. The Resource Realizer receives resource ids from the Device Manager and returns all instances available for this id along with their properties. The set of properties is based on the resource format. Continuing the example of David, the Resource Realizer returns the width, height, color-depth, file size, file types, and instance handles for the images. For the text files, it provides the file size, formatted content, such as tables and columns, instance handle, and file types. Just as a resource id provides access to set of related files, an instance handle provides access to a particular file that contains information about this resource. The Resource Realizer may either return the file itself (if it is locally available), a location pointer within the browser's file system (disk path) or a globally accessible location pointer such as a Web location.

The example above describes a situation where text, images, and videos share a resource id. A hypertext author could just as well create three resource ids, one each for instances separated by media type. She could then specify a caT hypertext where these resource ids were presented simultaneously, thus giving her viewers access to a small or large image, summarized or full text, and a short or a long video depending upon the characteristics of their devices. The Device Manager would decide independently which instance of each media format to present and the user may receive text summary on her phone, with a large image, and a high-definition video on her desktop computer.

While MIDAS enables authors to deliver content in multiple forms, there is authoring overhead involved in providing content in several forms. It is often a waste of a content creator's time to convert an image in forms that are suitable for presentation on mobile phones, tablets computers, and desktop computers. Such transitions, when these involve scaling of images to various sizes can easily be automated. Tools that support authors in converting information content to other media formats automatically or semi-automatically lower the overhead in creating multi-form content.

The Resource Manager [Park 2004] assists authors in adding resources to the resource repository. The Resource Manager works with a set of schemas that define relationships and the level of automation for converting information between various media types. For example, textual documents such as PDF or Postscript files may be automatically converted to plain text documents with some loss of formatting information; images can be scaled up or down in size or the number of colors they use. Similarly audio and video files may be downgraded to lower sampling rates in order to support the less capable devices. While some of these conversions may be completed automatically, others may require user intervention or post-processing. For example, scaling down a JPEG image from 320x240 pixels to 160x120 pixels can easily be

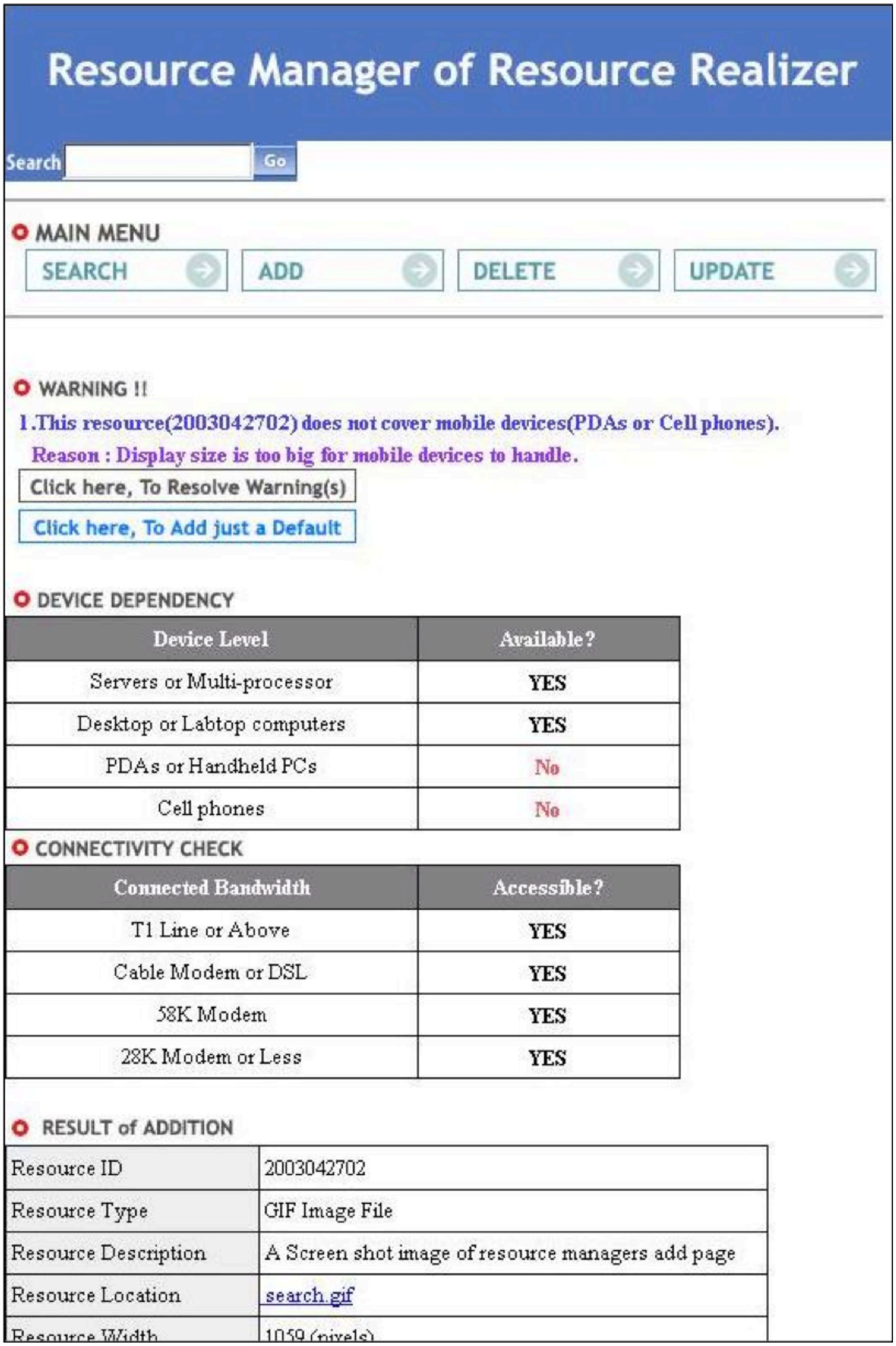

Fig. 6. Resource Manager

automated. However, tables or multi-column text in a PDF file converted to pure text may need to be validated by the author as such translations can have unexpected side effects. Conversion of information content is a complex process. While some users may be content to let MIDAS use its judgment in performing these conversions, others would surely like to be active decision-makers in deciding the fate of information content associated with their hypertexts.

While converting content to multiple forms for reaching the widest possible audience is an ideal but abstract goal. The Resource Manager connects the dots between content properties and those of the corresponding devices. Figure 6 illustrates a notification generated by the Resource Manager after a content creator has added a large image file. The image is too large to be displayed on desktop computers; however the file size is small enough that it can be presented over low bandwidth network connections. This interface allows users to perform the basic management operations on resource instantiations.

The resource repository employs a two-pronged approach to manage resources: the content files are stored in the file system by generating unique names for each file and the characteristics of the content are stored in a MySQL database. The metadata about the resource instances includes a pointer to the instance location. While this pointer could be a path within the local file system, it could just as well be a Web location.

Repository software, such as DSpace [DSpace 2011], enables repository managers to group related content files (called bitstreams) within a conceptual unit called item. Within an item, bitstreams may be grouped by subgroups called bundles. In the example of content related to David, the text descriptions may all be bundled together, as could the videos, and the images. Similarly, a Web document about David and images linked from this page may be bundled together. While items are analogous to MIDAS resources in that they both provide access to content in multiple forms, these representations differ in the level of metadata descriptions. DSpace associates metadata with items, treating multiple representations of an item similarly. Retrieving bitstreams that match specific criteria would require specialized programming. In contrast, the resource repository associates metadata primarily with the individual files, thus lowering the barrier to retrieve instances based on content properties.

5.5 BROWSER COORDINATOR

A user invokes the Browser Coordinator application on each device she wishes to include in her MIDAS space. This application is MIDAS' other extremity for communicating with caT. Just as the Device Manager acts as a client for caT servers, the Browser Coordinator acts as a local caT server for clients. It maintains the state of browsing for the particular device and translates from MIDAS' internal abstractions of content and actions to those used by caT.

When the Browser Coordinator starts, it registers with the Device Manager to indicate availability of the new device and sends over the device profile, which includes characteristics and user preferences for the device. Information access devices are characterized in terms of permanent and transient properties. Hardware and software capabilities such as display resolution, the number of colors they can render, processor power, and network bandwidth are all intrinsic characteristics of a device. Other properties may change more frequently: available storage space may decrease as a disk fills up, a user may install or uninstall media drivers or browsers, or enter a public place, such as a hotel lobby or an airport terminal, where playing audio may be disruptive to others.

The location of a device helps characterize its environment in terms of the degree of privacy a user has. This characteristic can help decide the modality (whether to play audio or not) or the level of detail to present. Interference indicates how distracted a user might be. For example, a user waiting in an airport lounge is in a public place but she may not be disturbed if she is traveling alone. On the other hand, a conference attendee has a higher potential to be engaged in conversation while she waits for her colleagues in the lobby of their hotel. GPS enhanced mobile devices can locate their position in geographic spaces, while larger devices may be situated in well-known positions.

Finally a user may opt to connect a device in one of three modes: presentation, input, or interactive. Presentation devices receive only content elements from the Device Manager. They do not receive action elements and, hence, do not permit users to interact with the content they render in any way. Input devices only receive actions from the Device Manager and not the contents; thus these devices present widgets that invite interaction from the user, for example links that a user may follow. Interactive devices receive both kinds of information elements.

The profile enables the Device Manager to route appropriate information elements to each available device. As shown in Figure 5, the Device Manager allocates content elements to devices by mapping the resource instances to the device characteristics, which are stored in the device registry. Upon receiving the information elements to be presented, the Browser Coordinator invokes appropriate browsers to render the content retrieved from the Resource Realizer.

5.6 BROWSERS

Browsers provide an interactive interface to MIDAS users. They render the information elements returned by the Resource Realizer. The browsers report user actions to the Browser Coordinator, which propagates them back to the Server. MIDAS devices may render one or more information elements. For example, a cell phone may deliver text, which is converted to audio form, while displaying images on the LCD screen.

caT browsers can interface with MIDAS through the Browser Coordinator without needing any modifications. Figure 7 displays the information content associated with Michelangelo's David in a virtual museum tour. The author has provided descriptive text, an overview of the sculpture, and a close-up of its face. In this MIDAS session, a caT browser is connected to the Browser Coordinator and is displaying content selected for a desktop computer. The browser also displays two links to the left of the text, which enable the user to navigate the museum tour.

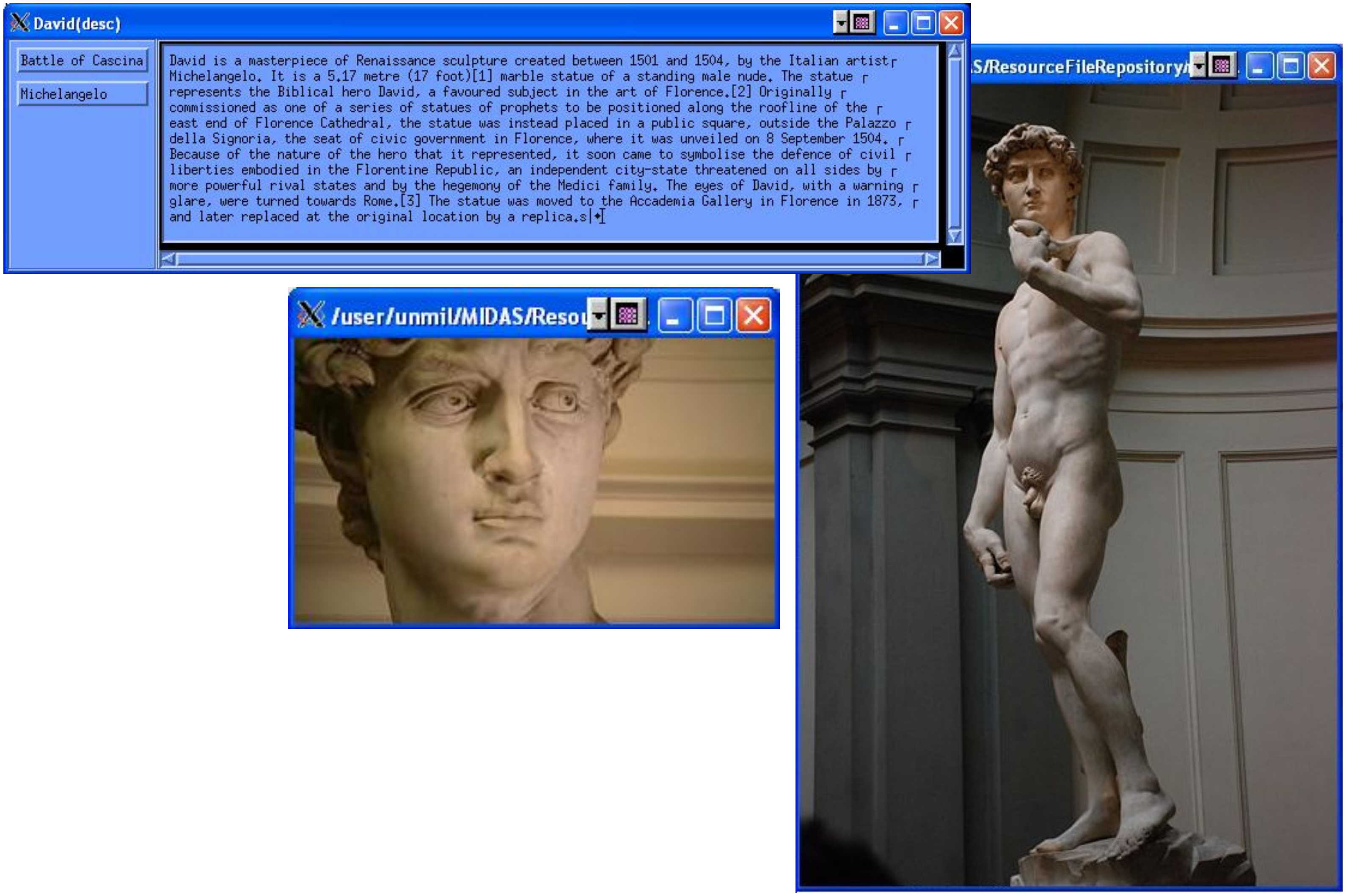

Fig. 7. Information about David in three caT browsers

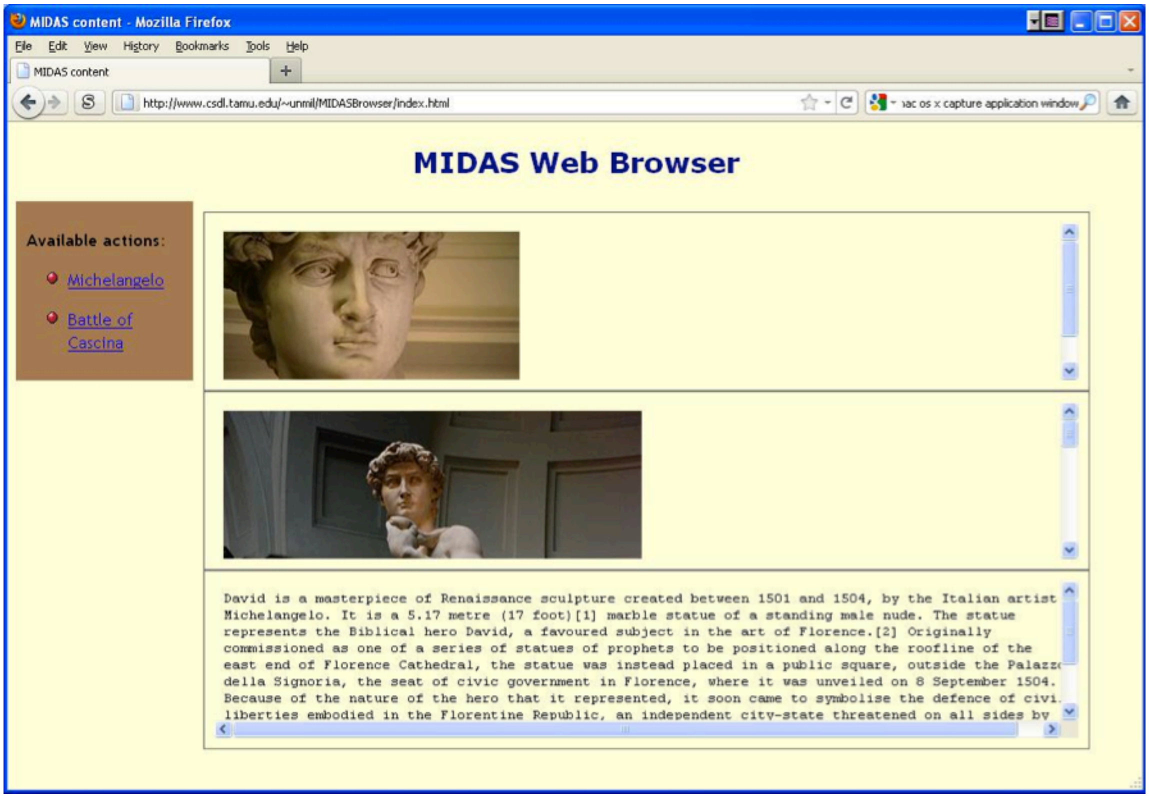

Fig. 8. Information about David in the MIDAS Web browser

5.6.1 MIDAS Web Browser

In addition to working with caT browsers, MIDAS enables developers to design custom browsers with little effort. I have developed a Web interface, functioning in conjunction with a Web server that a typical Web browser can interact with. The MIDAS Web browser receives updates from the Browser Coordinator, just as caT browsers do. Upon receiving a notification, the Web interface writes a self-updating HTML file to a known location. A Web browser can connect to this location to view the content generated by MIDAS. Figure 8 illustrates the information about David as displayed by the MIDAS Web interface. This page is constructed using a template that presents a two-column layout with available actions to the left and the content elements to the right. The MIDAS interface encodes the links to point to the User Action Listener,

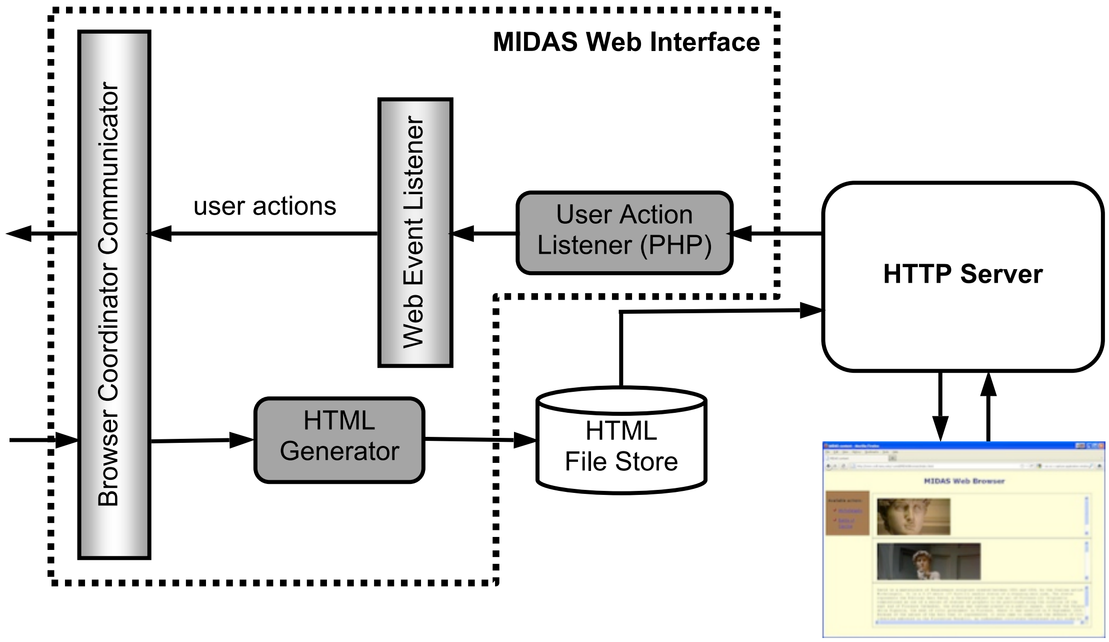

Fig. 9. MIDAS Web Browser architecture

a PHP script, that propagates the user's selection back to the MIDAS interface via a socket connection. Figure 9 presents the architecture of this interface.

Once the page is loaded in the Web browser, it auto-refreshes at a preset interval, currently set to 5 seconds. Due to lack of push functionality in the Web server, information updates propagated by the Web interface are displayed when the browser refreshes the HTML file. While not a perfect solution, this interface demonstrates the potential for MIDAS hypertexts interacting with users via Web browsers.

5.7 COMMUNICATION

MIDAS components communicate using connection-oriented, TCP socket-based connections. Communication is essential for registering and unregistering devices, transferring information elements among devices, propagating user actions, effecting state change updates received from the information service, and retrieving appropriate content forms. Figure 10 shows the communication paths between MIDAS components, identifying these with encircled numbers from 1 to 12, over the architecture illustrated in

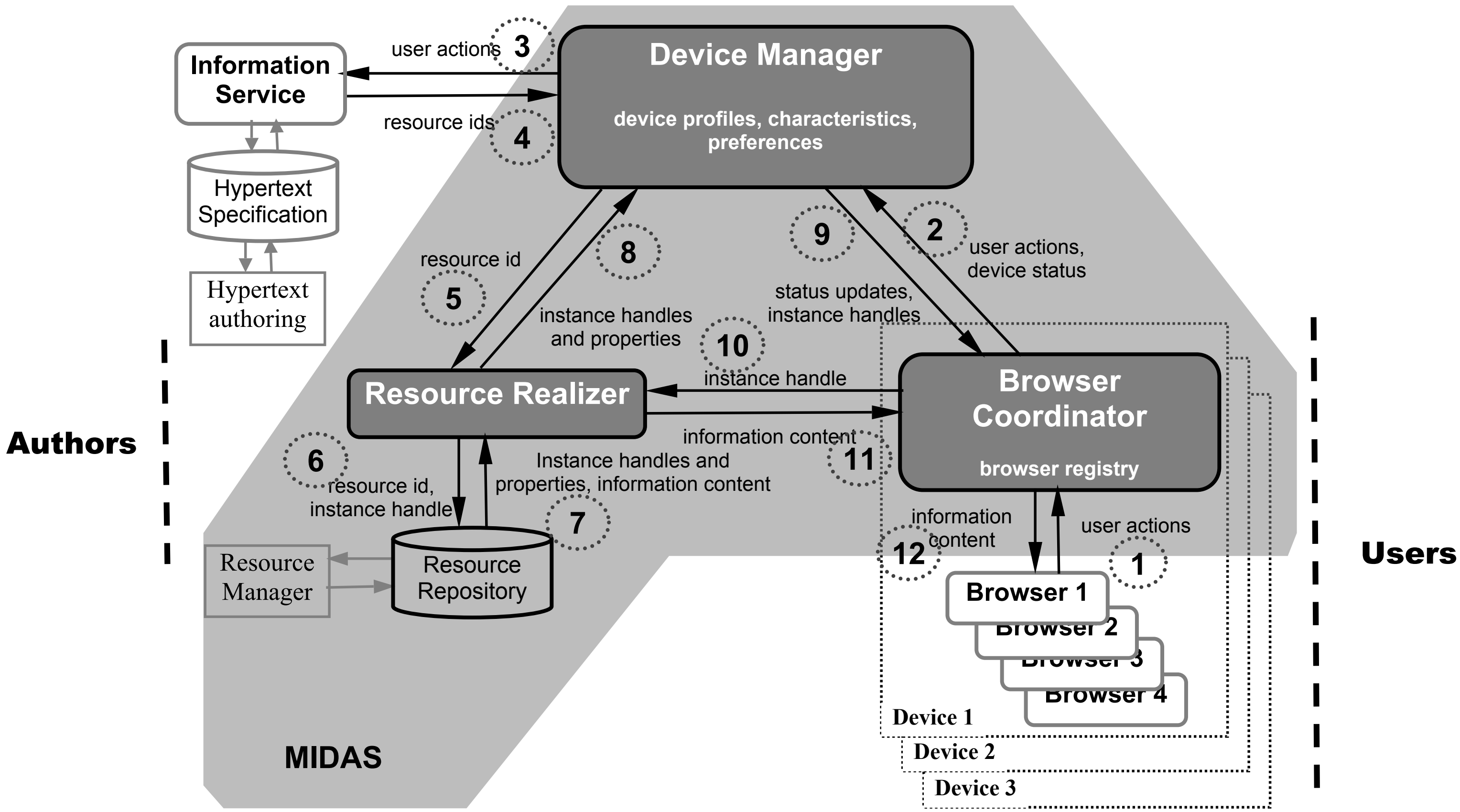

Fig. 10. Message paths in MIDAS

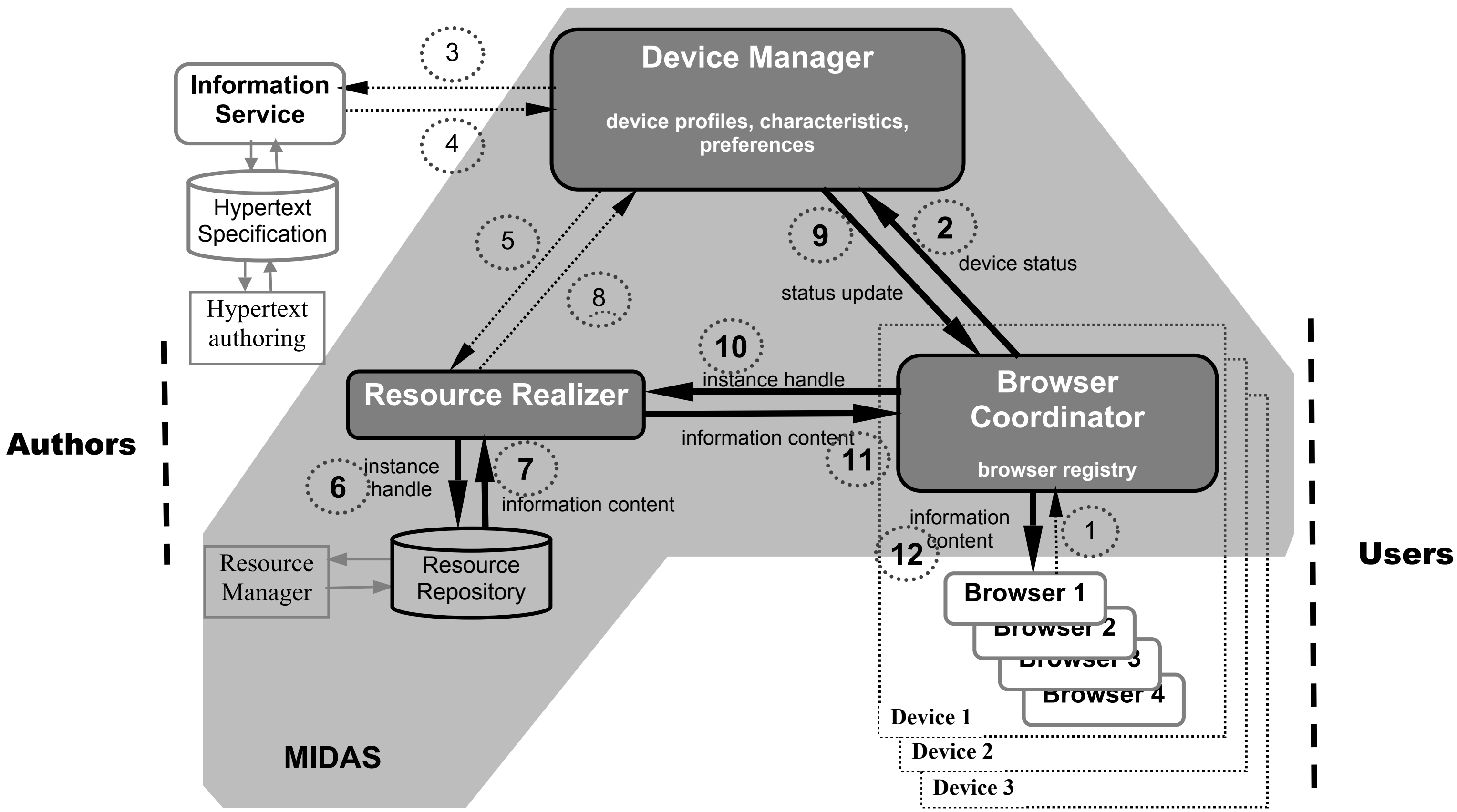

Fig. 11. Message paths activated when a device joins or leaves the MIDAS space

Figure 4. The paths are numbered in their order of activation when a user performs an action, initiating a change in MIDAS browsing state. User actions result in each path being used. The labels indicate the nature of the messages communicated along each path; some paths are used to convey different messages depending upon the situation. For example the Browser Communicator sends user actions as well as device status messages to the Device Manager along path 2. Not every event requires communication along all of the twelve paths. In general, events are initiated in one of three locations: browsers—when users click on links; information service—when state of the hypertext changes; or the Browser Coordinator—when a device joins or leaves the MIDAS space.

5.7.1 Device-initiated events

When the devices included in the MIDAS space change, the Device Manager reallocates the information content to ensure optimal rendering within the capabilities of the updated set of devices. The message paths followed when a device announces its intention to join or leave the space are highlighted in Figure 11. The Device Manager does not need to inform the information service regarding this change, nor does it request information about available instance properties, as it possesses these details. It responds to the initiating device, confirming that the request has been received and updates the list of available devices. It then reallocates the resource ids and forwards to each device the instance handles for rendering. The devices retrieve the content associated with the new instances and invoke the appropriate browsers.

5.7.2 User-initiated events

User actions, such as following a link, trigger changes to the state of the hypertext. The communication related to these events begins from the browsers and ends with the browsers displaying the updated contents. Commonly, this communication requires messaging along all of the twelve paths as the information service forwards new resource ids and the Device Manager must retrieve the properties and available instance handles for these resources. In some cases, these events may be triggered by change in a user's state when change in a user's state modifies the state of a hypertext. Some hypertexts may present mission-critical, secure items to local users only. When a user arrives in her office, her status changes from “traveling” to “local”, granting her access to additional content. In this case, the user does not interact with the hypertext at all in order to generate a new rendering; her reaching the office is sufficient to trigger a change in the presentation.

5.7.3 Information service-initiated events

Just as changes to a user’s state may affect the content available, so could changes to the environmental conditions. At the end of the workday, links for live help may become unavailable and content that encourages users to self-diagnose their problems, such as FAQs and knowledge bases, may be made available prominently. In such cases, the timing triggers the change to a hypertext’s state and such changes are unrelated to a user’s status. The end of the workday in the help desk employees’ time zone would affect users all over the world, regardless of their local time. Information service-initiated events trigger messages along paths 4 through 12, excluding only the initial forwarding of the user actions along paths 1 through 3.

5.7.4 Conflict management

Ensuring that MIDAS users perceive the information presented on different devices as a unified whole requires that all devices reflect a cohesive view of the current state of browsing. A user’s action of following links from two devices simultaneously could result in chaos absent a mechanism for resolving such situations. The Device Manager acts as the sole arbiter for resolving message conflict in MIDAS. Upon receiving a message that results in a state change, regardless of the triggering event, the Device Manager informs all connected Browser Coordinators to suspend user input. It then retrieves the properties of the updated resource set, allocates these resources to devices, and instructs the Browser Coordinators to present the new state before reactivating user input. While this solution is adequate for most scenarios, there remains a small time frame when messages may cross or a user event may be forwarded to the Device Manager before the Browser Coordinator has processed an instruction to suspend user input. In all such cases, the Device Manager ignores user input received after it has asked the Browser Coordinators to suspend input in order to maintain the consistency of the shared state across all devices.

The MIDAS architecture inserts a layer of indirection between the hypertext specification and the content that the browsers present to the users. This indirection enables coordinated channeling of content to devices with diverse characteristics as well as the adaptation of the presentation to the properties of the target device. The architecture achieves three of the six design goals outlined in Chapter I. It routes content to and actions from multiple devices and propagates the effects of these actions to other devices (goal #1). MIDAS relies on widely available technologies such as sockets and networking to enable multi-device information delivery (goal #2) placing few, if any, preconditions regarding the characteristics of devices that it serves. The Resource Manager includes support for automatic and semi-automatic transformation of information elements with a view to optimize content for rendering in different contexts (partial fulfillment of goal #3 as the transformation support is available at authoring time but not yet at presentation time). Finally, devices may join or leave at will. MIDAS reconfigures the presentation to render content optimally using forms that can be supported by available devices (goal #6).

The policies enacted in the Device Manager enable the architecture to meet the other three goals. These policies determine the rendered form of content elements as well as the target device that each element is rendered on. The next chapter describes how an understanding of human perception of images informs the process of transforming images for presentation on different devices. Chapter VII presents the content adaptation and device selection policies enacted by the Device Manager.

CHAPTER VI

HUMAN-CENTERED IMAGE TRANSFORMATION

In addition to accounting for device characteristics, MIDAS aims to design presentation and adaptation policies that account for human perception of information content. The improvements in the quality of digital cameras embedded in mobile phones has resulted in an upsurge of digital image capture as well as sharing via social media Web sites such as Facebook and Flickr. Typical mobile devices, such as iPhones, Blackberrys, and Android devices include three to five Megapixel cameras. Mobile applications for accessing these social media sites are easily available and popular. These applications enable users to upload their pictures to a service's Web site as well as to view photos that others have uploaded from devices with diverse form factors. While social media Web sites downsample the uploaded photographs for reducing storage and bandwidth costs, the pictures available to users easily overshoot the characteristics of mobile devices. Facebook limits photographs to 3 Megapixels (2048 X 1538 pixels) [Odio 2010]. Flickr allows free accounts to access images up to 0.8 Megapixels (1024 X 768), while those with pro accounts can view and download the original images, regardless of how large they are [Flickr 2011]. Phones, tablet computers, and ebook readers, are incapable of rendering these high-quality images in their original form and often scale down the images further for displaying within the display real estate available to the device. While several applications support interactive scaling of these images through multi-touch gestures, such as pinching and stretching, this approach displays only parts of the photograph at any time. Mobile devices typically take a one-size-fits-all approach, employing scaling as the only mechanism for fitting an image to the device's display screen.

I conducted a pilot user evaluation with a goal of assessing the user perception of similarity when size and color depth of an image changes, as these are the two primary attributes of device display screens [Karadkar et al. 2005]. The ultimate goal was to design image selection policies, informed by human perception of similarity, in order to present the best image from a human perspective.

6.1 APPROACH

The experiment was conducted in two stages. The first stage focused on gaining an insight into how humans categorize photographs. In this stage, five test subjects were asked to classify one hundred images. The subjects had the freedom to classify the images as they saw fit. Based on an analysis of these categories, I classified images into four types. In the second stage, I modified images of these four types by scaling and reducing their color depth. Five other test subjects viewed two forms of an image and rated their similarity. The subjects each completed a demographic questionnaire, performed the required task, and then answered freeform interview questions reflecting on their choices. Participants in stage 2 also filled in a post-task questionnaire. The following sections describe these steps in detail.

6.2 TEST SUBJECT CHARACTERISTICS

Ten test subjects participated in this study, five in each stage. The subject pool consisted of six males and four females between the ages of 18 and 45. The pool included individuals of varied ethnicities, nationalities, and academic backgrounds. The participants were all computer-savvy and had experience with using film as well as digital cameras. They uploaded images to computers and a couple of them posted their pictures on Flickr. None of the subjects claimed to be experts in creating on modifying digital photographs or images. They took pictures for their personal pleasure and to document their social lives and vacations. Most of the subjects used a cell phone and a few shared photos by phone via the multimedia messaging service (MMS).

6.3 CLASSIFICATION OF IMAGES (STAGE 1)

Participants in this stage were asked to classify a set of a hundred photographs. They were provided with no other instructions and had the freedom to organize their photographs in any manner they chose. The image corpus was generated by collecting

25 photographs each from four individuals. One of the contributions was short by one image, which I substituted by a 26th photograph from another donor. Each of the contributors was requested to provide a sampling of images that they had clicked and to provide images that were representative of their photo collection. The corpus consisted of photos taken during social occasions, on holidays, around the neighborhood, and at home. Subject matter pictured contained people (individuals and groups), animals, signs and notices, neighborhood improvement projects, bridges, buildings, natural vistas, and flowers, to name a few.

Each participant received a stack of 4” X 6” prints of these digital photos, thoroughly mixed (randomized) to eliminate the influence of prior subjects’ organization of these pictures. The subjects appreciated having physical copies that they could lay out on a large table and stack. The subjects received paper and pencil to create labels for their categories. While the suggested time for the activity was 45 minutes, subjects invested between 20 and 90 minutes into this activity and classified the pictures in 12 to 40 categories. Appendix A lists the categories created by each participant.

Most subjects were satisfied with their final set of categories; one admitted that she would not be satisfied with any categorization for long. The participants were enthusiastic and often went beyond the content visible in the frame by drawing on their experiences (“visiting a friend’s place”, “holiday photos”), knowledge (“London”, “texas”, “east asian country”), and opinions (“bad pictures”, “why?”, “artistic”, “delete”) to categorize the photographs. Some categories were very specific (“people-group-not aware of picture being taken”, “landscape without manmade – artistic”), some vague (“dude”, “same dude”), and yet others were general (“sport”, “scenic”). While any categories were based on visible content, some factored in color (“yellow and green”) and texture (“texture”). Subjects were unsure about how to classify some photos and they expressed this by uncertainty by creating descriptive categories (“Misc.”, “hmmm…”). While I checked in on the subjects periodically, I did not offer suggestions regarding the categorization, nor did I answer questions about the task before it was finished.

The subjects used an astonishing range of criteria for the classification. Since the ultimate goal of the study was to design policies for device-specific presentation, I emphasized content-based categories. From the 117 categories (only one was repeated—"food") provided by the users, I culled down the number to four, based on the dominant elements visible in each photographs. The categories are: those of individuals or groups ("people"), lettering or signage ("text"), animals or natural scenery ("nature"), and man-made objects ("structures").

6.4 EXPERIMENTAL SETUP (STAGE 2)

In this stage, the objective was to obtain numeric scores for the perceived similarity between images that have been modified for display on different devices. I selected four images of each type and converted them to the specifications shown in Table II, which matched those of commonly available devices such as mobile phones, ebook readers, tablet PCs, notebooks, and desktop computers. The images were scaled to the various sizes and to the different color depths. When modifying the color-depth of images, I included grayscale images as well. Thus, there were 24-bit grayscale as well as 24-bit full color images. These modified images formed the substrate for stage 2.

Participants in this stage viewed pairs of images that were modified along a single dimension. In order to avoid confounding the independent variables, each pair consisted of a picture in different sizes but the same color depth or in different color depths but at the same pixel size. The images were shown in separate browser windows on a dual-headed computer display, in order to ensure that the participants did not have

Table II. Image sizes and color depths used in stage 2

	Color depth	
Size	bits	(colors)
160 X 120	1	(2)
320 X 240	2	(4)
640 X 480	4	(16)
800 X 640	8	(256)
1024 X 768	24	(16 million)

to deal with the cognitive overhead of distraction due to switching application windows when comparing the images. The size and color properties of each image were displayed. Each subject viewed 20 randomly generated pairs that differed in size and 20 that differed in the number of colors. The study was not timed; subjects could view each pair for as long as they wished before answering three questions: how similar they thought the images were (on a 9-point semantic differential scale), whether they would allow a computer program to replace the more expressive image with the less expressive one (automatically vs. manual choice), and how likely they were to permit such replacement if they were in-charge of approving the change (on a 9-point semantic differential scale).

6.5 DATA ANALYSIS

When images shrink in size or lose color, information is lost. The amount of information lost can be calculated in as many ways as the quantity of information

Table III. Content lost when images are scaled

Size	# bits (size)	% content retained	% content loss
1024 X 768*	786,432	100	0
800 X 640	512,000	65	35
640 X 480	307,200	39	61
320 X 240	76,800	10	90
160 X 120	19,200	2	98

Table IV. Content lost when images are color-reduced

# colors	# bits (color)	% content retained	% content loss
16 Million	24*	100	0
256	8	33	67
16	4	17	83
4	2	8	92
2	1	4	96

*Value with the most information content in an image content is measured. For example, a 1024 X 768 pixel JPEG file may require about 180 KB for disk storage. Reducing this file to 160 X 120 pixels results in a disk size on the order of 10 KB. The information lost in this case could be represented by the 170 KB of disk space freed. As the JPEG file format employs compression, the relationship between the disk space saved and the amount of information content lost may not be directly proportional. More meaningful metrics of information loss with regard to my purpose are the bits lost due to reduction in pixel count and the reduction in color count. Table III shows the information content lost when an image is scaled in size. An image consisting of 160 X 120 pixels retains only about 2% of the bits contained in an image that consists of 1024 X 768 pixels. Similarly, an image that encodes color information in 1-bit contains about 4% of the information contained in a 24-bit image, assuming that no other attributes (such as size) change.

The calculations in Tables III and IV use the properties of the image with the highest resolution as the baseline. Users in our study viewed and compared image pairs with different base properties: for example they compared 640 X 480 pixel images to 160 X 120 images as some devices then included VGA cameras. As the scales shown in Tables III and IV are not linear, the information content lost changes with the baseline used for comparison. Thus, a more appropriate baseline to use for analyzing the data is the actual information content loss in the images that the subjects viewed. These values are shown in Table V. The distance in this table refers to the difference in image properties shown in Tables III and IV. A 160 X 120 pixel image is at a distance of one

Table V. Percentage of content loss in images viewed by test subjects

distance	scaling	color reduction
0*	-	67
1	56	54
2	80	79
3	93	90
4	98	96

*Only applicable when a color image is converted to grayscale without changing the number of bits from one with 320 X 240 pixels and at a distance of three from one that has 800 X 600 pixels. Similarly, an image that represents colors in four bits is at a distance of one from those that use 8-bit color as well as those that use 2-bit color. A key point highlighted in Table V is that the amount of information content loss when images are scaled or color-reduced is numerically similar, with more content being lost when images are scaled than when color content is reduced.

The following subsections document and analyze the participants' responses. While the responses do not yield statistically significant results, they point to trends and provide an insight into the subjects' perception of similarity when image properties were modified.

6.5.1 Perception of image similarity

Table VI shows the effects of scaling and color depth reduction on subjects' perception of similarity. On the 9-point scale used, the scores decrease with increasing distance, indicating that as images reduced in size or colors, the subjects' perception of similarity decreased. The scores decreased for both kinds of modifications but not to the

Table VI. Subjects' perception of image similarity

distance	scaling	color reduction
0		6.89
1	7.12	6.03
2	7.11	4.57
3	6.87	4.93
4	6.14	4.14

Table VII. Scaled image similarity by type

distance	people	text	nature	structures
1	7.17	6.78	6.57	7.67
2	7.09	7.25	7.33	6.78
3	7.00	6.29	7.75	7.00
4	5.71	5.00	7.00	7.50

same extent. The perception of similarity for color-reduced images diminished sharper than that for scaled images. This change in scale becomes more pronounced as the distance between the images in a pair increases. While the numbers in the columns are expected to decrease with increased distance, the numbers in a row are expected to be closer to each other as the content loss is similar numerically for scaling and color reduction. In fact, Table V shows that marginally more content is lost when an image is scaled down than when it loses colors but the subjects' perception of similarity declined more for color change than for scaling.

A closer look at the scaled images reveals that the subjects perceived some types of images to retain their similarity better than others, especially in the case of extreme scaling, when over 90% of the content is lost. In Table VII, the scores are remarkable lower for image pairs of people and text at a distance of 4. In the freeform interview, subjects expressed disappointment that they could not observe the faces of people in the smaller images or read the text. That people are interested in other people is a result that studies related to the behavior of people in online communities have long known [Girgensohn and Lee 2002]. Images of natural elements and manmade objects scale more gracefully. In contrast, color reduction affected all types of images similarly with no discernable patterns.

The data for similarity between grayscale images and color images, shown in Table VIII, indicates that subjects perceived color images that lost content to be closer to their original counterparts than they did for grayscale images. The subjects' responses for perceived similarity between pairs of color images are higher than those for the perceived similarity between pairs of grayscale images. They also perceived that the loss

Table VIII. Perception of similarity for color and grayscale images

distance	within color	within grayscale	color to grayscale
0	-	-	6.89
1	6.88	6.50	5.57
2	5.25	4.17	4.55
3	6.20	5.00	4.00
4	-	4.34	4.00

of content was more significant when color images were converted to grayscale, as evidenced by lower similarity ratings in a given row. Intuitively, this makes sense, as removal of the color information in addition to reduction in color depth is a greater loss of content than either of these independently. In the top row, a distance of zero is used for image pairs that included a color image and a grayscale image with the same number of bits—or pairs of images with the same color depth. As expected, the subjects found these pairs to be the most similar. In Table VIII, a distance of four for within color images is not a valid case as this distance only applies when a 24-bit image is color-reduced to a 1-bit image, in other words, a bi-tonal, black-and-white image.

6.5.2 Image replacement

In addition to commenting on the similarity of images, subjects also indicated their willingness to have a poorer image replace a more expressive one. They responded whether they would accept automatic substitution or would expect a program to request permission from them before displaying an altered image. In response to the final question, they indicated on a 9-point scale, how likely they were to permit this

Table IX. Subjects' responses for scaled image replacement

distance	viewer permission for replacement	automatic replacement
1	6.05 (56%)	7.66 (44%)
2	6.00 (51%)	7.42 (49%)
3	6.50 (50%)	7.13 (50%)
4	4.18 (79%)	7.67 (21%)

Table X. Subjects' responses for color-reduced image replacement

distance	viewer permission for replacement	automatic replacement
0	5.10 (56%)	7.50 (44%)
1	4.44 (64%)	7.43 (36%)
2	3.89 (90%)	8.00 (10%)
3	4.00 (71%)	8.75 (29%)
4	3.80 (71%)	8.00 (29%)

substitution if a program informed them that they would be viewing a less-than-optimal image. Tables IX and X reflect the subjects' attitudes toward such image replacement. The percentage of cases in which subjects would prefer automatic vs. manual substitution is presented in parentheses, next to each score. As there are only two options, the percentages in each row add up to hundred. While the subjects were expected to answer the final question only if they disapproved of automatic substitution, all subjects answered this question, providing more data than expected.

As Table IX illustrates, the participants would accept automatic image substitution in about half the cases up to a distance of 3. They also indicated a higher likelihood of permitting this substitution if the program asked them, as evidenced by higher numeric scores. When the images were scaled to an extreme, however, their willingness to accept automatic substitution diminished and less than a quarter of image pairs passed muster. However, for these images, the subjects were just as confident of permitting a substitution if asked by a program. Conversely, they were more conservative in permitting substitution of images that they were not comfortable with replacing, as indicated by a drop in the score to 4.18, which is in stark contrast to the other scores in this table.

Table X reaffirms that subjects viewed color as a more critical attribute of images than size. They were more conservative about accepting automatic replacement of these changes, especially after more than 1-step distance and are more certain that they will permit substitution for this smaller percentage of images. Similarly, they were more reluctant to permit replacements to images, reflecting this approach with lower scores. The highest score for replacement (5.10) as well as the highest willingness to accept automatic substitution (44%) were reserved for cases where color was lost but not the color depth. One of the subjects candidly stated that he would never allow automatic substitution for any case where the image quality changed, whether consequential or not. Another participant was comfortable with automatic substitutions only for color-to-grayscale changes but not in the case of loss of color depth or resolution.

6.5.3 Post-task questionnaire

The Web interface used for conducting the study presented the screen size and color depth for each image. Some subjects did not realize that this information was displayed, while others used it to validate their impressions formed by a visual inspection of the image pair. One of the subjects was willing to accept “minor” modifications to images, regardless of their form while another was more likely to accept changes to image resolution than to the color depth.

6.6 APPLICATION OF RESULTS

A high-level goal of this study was to design policies for displaying images effectively within the constraints of the devices that are available to a MIDAS user. From this perspective, the study yielded the following key insights:

- Color is perceived to be more critical for retaining integrity of images. When possible, prefer scaling to color reduction.
- Not all images scale equally gracefully. Photos of natural vistas and manmade structures scale better than those that show recognizable people or text.
- Displaying images in grayscale works best when it is not coupled with reduction in the number of bits.

These results can be applied to develop human-centered policies for selecting optimal images to display in MIDAS hypertexts. The study used a corpus of modified images to simulate changes to an image when it is displayed on different devices. In practice, the only differences between an image and its display properties may be caused by properties of the target device.

While the prospect of automatically applying different policies based on image content seems challenging at first sight, the characteristics that the study participants cared about have both been of keen interest to image recognition investigators. Both face and text recognition in images have emerged as significant research threads. Hjelmås and Low [2001] present an excellent outline of approaches for face recognition.

Similarly, text recognition approaches have been studied extensively for over a decade [Kim et al. 2003; Wu et al. 1999; Ye et al. 2005]. Thus, MIDAS' resource repository could be augmented to classify photographs automatically upon their addition. While the images depicting nature and manmade objects could not be easily separated, such distinction may not be necessary as identical policies apply to both types of photos.

These results displaying grayscale images should inform the design of policies for presenting images on e-ink-based readers, such as the Kindle [Amazon 2011] and the nook [Barnes & Noble 2011], both of which are equipped with a 4-bit grayscale display. In particular, the underlying Android operating system makes the nook an attractive option for designing custom applications, such as MIDAS' Browser Coordinator.

In the following chapter, I will discuss the development of human-centered image presentation policies grounded in the results of this study.

CHAPTER VII

CONTENT PRESENTATION

The content presentation policies operationalize MIDAS' ability to decouple content associated with a hypertext from the specific form in which it is rendered. The Device Manager resolves the resource ids received from the information service via interaction with the Resource Realizer. It retrieves the available content forms for each active resource as well as the properties of active devices and applies presentation policies that map the resources to devices as well as select the specific resource instances to render. The policies are applied at two interacting levels: instance scoring and device selection. In order to implement these policies, MIDAS externalizes the characteristics of devices as well as content for ease of access at presentation time.

7.1 CONTENT AND DEVICE PROPERTIES

The Resource Realizer maintains the content instances and their properties in a resource repository. The repository stores instance properties in a database along with

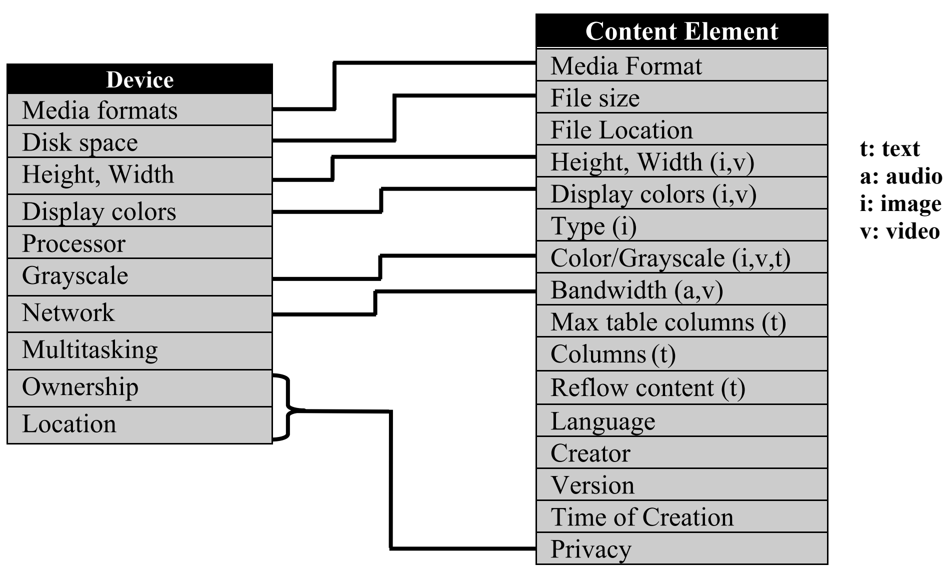

Fig. 12. Mapping of device characteristics to content properties

pointers to files, which are stored in a configurable, designated part of the file system. The database externalizes resource properties such as the file size and media format. In addition, as outlined in Figure 12, the repository stores media-specific information such as height, width, and number of colors for images as well as videos. Images are classified into one of four categories based on the dominant elements visible: those of individuals or groups (people), lettering or signage (text), animals or natural scenery (nature), and man-made objects (structures). Whether the content instances consist of color or grayscale elements is also recorded. For textual content, the repository stores details regarding formatting content such as columns in text files and the most columns in tables. While some textual encodings, such as HTML or pure text, allow reflowing the text, pdf files do not afford this flexibility. The repository also stores the bandwidth recommended for effective streaming of audio and video files. Much of this information, if not all, can be extracted programmatically from the content files, when these files are added to the repository, thus minimizing the overhead on authors' time. The database schema also supports the storage of descriptive and administrative metadata, such as information about the creator, time of creation, version, and language, and privacy settings under which the content may be presented.

The device profile consists of the display characteristics (height, width, and color depth, and the ability to show color or grayscale), disk space available for MIDAS (the local cache space), media formats that a device can present, the speed of the network connection, whether the device can present multiple content elements simultaneously (many mobile devices do not support user-level multi-tasking), whether a device is entirely controlled by the user or is under shared control (for example a public device that is available temporarily) and the location of the device, which indicates the degree of privacy available to the user.

During a browsing session, content attributes are compared to those of the target device and the best scoring content element is selected for presentation. In some cases, comparing the attributes is a straightforward process. For example, the media format of an element to render must belong to those supported by the target device; if a device cannot play audio files, all audio content is automatically excluded from consideration. Similarly, files that are larger than the MIDAS disk cache or those that require more bandwidth than available on the device are not considered.

The remaining instances are all viable candidates for presentation on the target device. Some may be a better match for the device than others. The next step evaluates the suitability of each instance using a media-specific cost function to select the most appropriate one.

7.2 CALCULATION OF INSTANCE SCORES

The Device Manager treats all instances in the resource repository—including those modified by the Resource Manager—as unmodified instances and assumes that these have been vetted to ensure their integrity to the author's satisfaction. In calculating the instance scores, the only question that the Device Manager addresses is the modification that would be introduced in an instance due to the difference between instance and device characteristics.

The instance scores calculated use form-dependent mechanisms. For some media, such as audio and video, the scores are calculated based on primitive properties that directly correspond to those of the available target devices, for example display space and bandwidth. These scores do not account for more involved attributes, such as the content and its presentation. In contrast, scores for images and textual documents account for content properties and a conception of how these properties affect user perception.

Each instance retrieved from the repository is assigned a score of zero. Thus, the instance that scores closest to zero after accounting for its presentation characteristics on a target device is the best fit for the device.

7.2.1 Image score

Images are scored based on up to four parameters: the kind of conversion being effected: scaling or color reduction; the extent to which the image is being subjected to this conversion; whether the image is being converted from color to grayscale; and the presence of people or textual elements in the visible content when scaling images. The score is calculated using the equation below:

$$W_{scaling} \times \left(\frac{9 - W_{scaling,steps}}{9} \right) \times W_{type} + W_{color} \times \left(\frac{9 - W_{color,steps}}{9} \right) + W_{togray} \times \left(\frac{9 - W_{grayscale,steps}}{9} \right)$$

$W_{scaling}$ = 0.4	W_{type} = 0.53 (people, text)	W_{togray} = 1.0 (color image)
W_{color} = 0.6	W_{type} = 0.47 (structures, nature)	W_{togray} = 0.0 (grayscale image)

This formula allows three different possible conversions being applied to an image by virtue of disparity between image and device characteristics. Each of the transformations that is applied increases the numeric score of the resulting representation, making it less desirable and less likely to be selected.

The formula favors scaling over color reduction by allocating a lower modifier (0.4 against 0.6 for color reduction). The numbers reflect the ratios of subjects' response for similarity measures in relation to each modification. Similarly, the type of the image affects the scaling, allocating a higher weight to scaling of images with people and text, thus weighing against scaling these images significantly. The term $W_{scaling,steps}$ applies a weight equivalent to that shown in Table VI in response to the corresponding amount of content loss as shown in Figure 5. This term weighs 9.0 for an image that is not scaled (no content loss) thus contributing nothing to the final score, which increases the desirability of the instance. Similarly, an image that is scaled to 20% loses 80% of its content, corresponding to the third step. This results in a weighing factor of 6.87 being used for scaling this image. As this example illustrates, the formula uses thresholds for calculating the amount of information loss rather than rounding as a content loss of 80% is closer to that for 2 steps in Table V (79.56%) rather than that for 3 steps (93.11%).

The values for weighing the instance score due to color reduction as well as grayscale presentation of color images have been similarly derived. When the original image is a grayscale image the last term contributes nothing to the final instance score (W_{togray} = 0.0), as grayscale display of a grayscale image constitutes no material change.

7.2.2 Text score

The score for text files is calculated based on the width necessary to display the file, the potential for losing color from the file and the size of the file relative to that available on the disk. The following formula expresses the interplay between these factors:

$$\frac{\left(W_{reflow} \times Columns_{text}\right) + Columns_{table}}{Wd_{device}} + \frac{Gr_{device} \times Text_{togray}}{W_{grayscale}} + \frac{Disk_{text}}{Disk_{device}}$$

$Columns_{text}$: text columns	*Gr_{device}: grayscale display*	$W_{grayscale}$ = 50.0
$Columns_{table}$: most table columns	W_{reflow}= 0 if flexible	*Disk: disk space*
Wd_{device}: display width in pixels	layout, 1 otherwise.	*$Text_{togray}$: Color content*

The horizontal screen space necessary to display a file is dependent upon the contents of the file such as its columnar layout and the width required to display structured information, such as tables. HTML files and text files include fewer constraints on altering the layout than do presentation-ready formats like pdf or Microsoft Word (doc, docx). The freedom to reflow a document relaxes the constraints on columnar layout but does not affect the presentation of tables, as columns are semantically related to each other. Thus, the first term in the formula accounts for the necessary width, attributing better scores to files that support more flexible presentation or to files that are presented on larger displays. The second term indicates the potential for information loss due to conversion from color to grayscale. The term only weighs in when files with color elements are scored for grayscale displays. Gr_{device} has a value of 1.0 for grayscale devices and 0.0 otherwise, while $Text_{togray}$ has a value of 1.0 for color content and 0.0 otherwise. Thus, this term evaluates to a non-zero value only when color content would be displayed on a grayscale device. In this situation, $W_{grayscale}$ acts as a normalizing factor to avoid placing an unduly high emphasis on the grayscale conversion. Finally, the third term favors files with lower disk space utilization ratios. In the case of devices with large disk caches, the file size plays a less critical role in the selection process. However, on space-constrained devices, this term impacts the decision to select smaller files significantly as lower disk cache results in higher values for this term.

7.2.3 Video score

The selection of video and audio files is dependent solely on their display and streaming properties. The display properties include the height and width of the frames

$$\frac{|Ht_{device} - Ht_{video}|}{Ht_{device}} + \frac{|Wd_{device} - Wd_{video}|}{Wd_{device}} + \frac{|C_{device} - C_{video}|}{C_{device}} + |Gr_{device} - Gr_{video}| + \frac{BW_{video}}{BW_{device}}$$

Ht: height	*C: color depth*	*Gr: grayscale*
Wd: width	*BW: bandwidth*	

and their color depth in bits. Best scoring videos undergo minimal changes to these properties for presentation on the target device. Color videos that are presented on grayscale displays are scored lower due to loss of information. A net negative score does not constitute a problematic score as the best scores are defined as those that are closest to absolute zero. The formula, however, guards against the possibility of the score being artificially lowered in the case of a few terms evaluating to negative values while others evaluate to a positive value. Finally, the score includes a measure of bandwidth usage; instances that would dominate the available bandwidth are scored lower.

7.2.4 Audio score

Since audio files have no visible interface the bandwidth usage is the sole, simplistic metric used for selecting audio files as presented below:

$$\frac{BW_{audio}}{BW_{device}}, BW : bandwidth$$

7.2.5 Score-based instance selection

After eliminating the non-viable instances, the Device Manager calculates the score for each remaining instance for each available device. It then maps the best-scoring instances to the devices based on a device selection policy.

7.3 DEVICE SELECTION POLICIES

The Device Manager implements several policies to map content to devices. The simplest policy mimics caT by channeling each information element to each device. Other policies are more involved and some direct instances to the devices without replication. A user selects the policy that MIDAS applies for a browsing session. The Device Manager reads the policy via a configuration file when it starts. The chosen policy remains effective for that invocation of the Device Manager, which, for all intents and purposes, corresponds to a MIDAS browsing session.

7.3.1 Full replication

When a user selects the full replication policy, the Device Manager selects the best instantiation of each content and action element for rendition on every available device. This mode of operation is similar to the behavior of the caT server, which treats all clients identically and forwards every active content element and action to each client. The only difference between the modes is that while the caT server sends identical content (the same file) to each connected client, the Device Manager sends the best instance of a resource for rendition on the target device.

7.3.2 Interaction mode-based replication

The interaction mode-based replication allows each connecting device to specify whether it is an input, output, or interactive device. The Device Manager then forwards instances of every active content element to all output devices and instances of every active action element to all input devices. Interactive devices belong to sets of both input and output devices. Thus, they receive contents as well as actions. This policy allows devices to state their preferences and serves as the base policy for implementing others that are described below. In other words, both content optimal and MIDAS optimal policies respect a device's interaction mode. Both these policies select appropriate content for presentation on devices but replicate the action elements on all input devices in order to present users with the most flexibility in browsing.

7.3.3 Content optimal

In the content optimal mode, the content is not replicated to every output device but a content element is sent only to the device that can best render it regardless of the form. To accomplish this, the Device Manager scans the scores of all instances for a resource and selects the lowest scoring resource. It then channels this instance to the device that it scored the lowest on. Thus, each resource is rendered only once. It is possible that a powerful device, such as a desktop computer may overshadow other devices by returning the best score for all resource instantiations. This may result in starvation of other output devices, which will receive none of the active content elements. The content optimal policy does not prioritize the use of multiple devices that are available during the browsing session.

7.3.4 MIDAS optimal

The MIDAS optimal policy balances content optimization and device co-use. It calculates the best score for the presentation of all active content elements without regard to optimizing the score for individual instances. Thus, the solution found by this policy may sacrifice the best instances in favor of the best rendering for all instances. This policy takes a brute force approach, calculating the scores for all possible presentation of content elements. This approach is not scalable and may result in unacceptable delays in finding the optimal solution for a large number of instances related to many resources that are being considered for presentation on several devices. For example ten resources with ten instances each being considered for ten devices would result in one thousand scores. While the possibilities increase exponentially, it works well for "realistic" situations, where a user has access to two or three devices.

MIDAS' content selection policies include a human perspective when presenting images (partial fulfillment of goal#4) and render information on devices that are most suited for the instances being presented (goal #5).

The next chapter evaluates the MIDAS architecture and policies in the context of the 12C framework, comparing it to other multi-device interface systems.

CHAPTER VIII

MIDAS: A 12C ANALYSIS

MIDAS' architecture and policies together instantiate a configurable multi-device interface system that tailors its presentation to both characteristics of available devices and human perceptual abilities. Mirroring caT's approach to explore the articulation points in hypermedia by separating the specification of its components [Furuta 2005], MIDAS configurations are flexible, separating the specification of different aspects of multi-device interfaces. Building on caT's approach of granting autonomy to clients (browsers), MIDAS components each perform specific tasks and have great flexibility in fulfilling their roles. The Device Manager and the Browser Coordinators jointly specify different aspects of MIDAS' behavior. This flexibility enables MIDAS to serve as a vehicle for exploring various interaction schemes between components in a multi-device interactive interface. In the following sections, I analyze MIDAS along the dimensions of the 12C framework and compare it to other multi-device software. Table XI updates Table I to include MIDAS and repositions caT next to MIDAS for facilitating comparison between the two systems.

8.1 CONCURRENCY

As a multi-device system, MIDAS renders information on more than one device at a time. While a user may have multiple devices available to her, she retains control over which of these she wishes to include in the MIDAS device space. A decision to include or exclude a certain device is flexible, as she only has to start or stop a Browser Coordinator as a local application on the intended device. A user could further minimize the effort involved in including devices in the space interactively by configuring the Browser Coordinator when the device starts up. Of course, she still retains the ability to remove it from the space when she so chooses. With some support from the underlying operating system, MIDAS can thus support a hybrid mode where some devices may be included automatically when they are available and a user may add others interactively.

Table XI. Characterization of MIDAS in the 12C framework

12C Attribute	MIDAS	context-aware Trellis	WebSplitter	Multibrowsing	Ubiquitous Display System	ANMoLE	IDD	Pebbles
Concurrency	manual, automatic (OS) public, local, remote	automatic personal	automatic personal	manual public	manual public (2 devices)	automatic personal	manual, automatic personal	manual personal
Control	centralized, distributed (device mode-dependent)	distributed	distributed	distributed	centralized	?	central, distributed (strategy-dependent)	application-dependent
Comity	unrestricted	unrestricted	unrestricted	unrestricted	large displays	unrestricted	unrestricted	Windows
Completeness	Y (but not of specific instantiations)	browser-dependent	?	Y	Y	N	N	Y
Coverage	Flexible (Dev. Mgr. policy-dependent)	N	Y	N/A	N/A	Y	Y	N/A
Conversion	Y (authoring, device)	N	N	N	N	Y	N	N
Composition	generative	generative	filtered	generative	generative	generative	filtered	N/A
Coherence	Y (may be influenced by Dev. Mgr. policy)	browser-dependent	Y	Y	Y	Y	?	Y
Coordination	User, Device, Environment	User, Environment	User	N/A	N/A	User	User, Device	User
Continuity	Y	Y	?	Y	Y	?	Y	Y
Constancy	N	Y	Y	Y	Y	N	Y	Y
Confidence	N	N	N	N	Y	Y	N	N

Systems such as WebSplitter [Han et al. 2000] and ANMoLE [Haneef and Ganz 2004] automatically include all the devices available to a user in their presentation scheme. This approach leaves little room for addressing situations where devices available to a user may need to be reserved for tasks that are unrelated to the multi-device presentation. For example, a user may not wish to channel content to her work phone because it must be dedicated for business-related activity.

MIDAS automatically channels information to all the devices that are included in the space. Unlike the multibrowsing framework, the creator of information need take no special precautions for facilitating information presentation on multiple devices. Also, unlike the Ubiquitous Display System [Aizawa et al. 2002], the viewer is not required to forward content to particular devices interactively. The Device Manager's policies address both the channeling of the information to specific devices as well as the presentation of content in a format that is suited for the target device, without necessitating interactive input from the user. Thus, MIDAS is developed with an intent to minimize the cognitive load on information creators as well as consumers.

MIDAS makes few assumptions about the properties of connecting devices. Any device that runs the Browser Coordinator is a MIDAS client, be it a personal device or a public one such as a large-screen display. Devices identify both their ownership and their location as shown in Figure 12. These two aspects together determine the degree of privacy available to a user. A public display located in a user's hotel room affords the user more privacy than an identically configured public display located in a hotel lobby. The Device Manager can include privacy metrics as a factor in its presentation policies when determining the appropriateness of content that is directed to an appliance.

8.2 CONTROL

Multi-device interface systems typically support a single, pre-defined control strategy that is binding on each device that participates in the presentation. The multibrowsing framework, WebSplitter, and context-aware Trellis all support distributed control. The information server propagates actionable elements to multiple client devices and listens for user events from these clients. Not all WebSplitter clients may receive control elements as the content routing is guided by system and device policies. Similarly, Interface Distribution Daemon (IDD) selects a strategy when the system is initiated, which remains in force for the duration of the session [Luyten and Coninx 2005]. Some IDD policies allow distributed control while others do not. The Pebbles system [Myers et al. 2000a; Myers et al. 2000b] selects application-based control strategies. While applications that use mobile devices to control presentations use centralized control, those that extend the desktop display to a mobile device may allow for distributed control. In each of these settings, the decision regarding control strategy is pre-defined, based on application, session, or system properties.

In contrast, MIDAS allows users to specify how they wish each device to interact with the information service. When a device joins the space, the Browser Coordinator conveys the interaction mode (input, output, or interactive) to the Device Manager, which always respects the user's choice. It channels control (interaction) elements only to input and interactive devices. Thus, MIDAS does not define an interaction strategy at all. Strategies evolve based on the preferences of the devices included in the space at any time. Depending upon the modes specified by various devices it is possible for a MIDAS session to start with centralized control, switch to distributed control as additional input devices become available, and switch back to a centralized control. Indeed, a session may allow for no user control at all, if the user has requested that every device in the space connect in the output mode. However, a lack of controlling devices does not imply a static, unchanging browsing state. WWW documents as well as caT documents may change in the absence of user interaction. While dynamically generated, self-refreshing WWW documents may change their content due to specialized programming, caT hypertexts can "browse" independently of user action, based on the activation conditions embedded within the hypertext structure [Furuta and Na 2002].

8.3 COMITY

In an attempt to support devices with diverse characteristics, MIDAS places few restrictions on devices that may join a space. MIDAS devices are primarily defined by the fact that they run a correctly configured Browser Coordinator. Browser Coordinators could be developed for a variety of operating systems such as Linux, Windows, iOS, and Android, to include these devices in a MIDAS space. The device profile defined by the Browser Coordinator expresses the relevant characteristics of this device to the Device Manager. For example, the device profile currently includes display characteristics of the device but does not include information about its camera. The device profile may be expanded to include camera information, gestural interaction support, and any other relevant attribute of devices that MIDAS hypertexts may desire access to. Maintaining a comprehensive profile is a moving target as appliance manufacturers pack more features with each generation of devices. The newest devices include GPS locators, magnetic compass, accelerometers, and gyroscopes.

MIDAS also does not require devices to be located proximal to the user. In the scenario described in Chapter I, Eva could just as well have added her mobile phone to her MIDAS space without having her home computer leave it. MIDAS would then present information on the two devices, although no one at Eva’s apartment would access the information being presented there. Instead, if Eva had included her home printer in her browsing session, an appropriate browser could print the content that is sent to this device for later perusal.

Like MIDAS, many multi-device systems strive to reduce property-based requirements with a view to support a broad set of devices. A few early systems, such as Pebbles, were developed for the MS-Windows platform but the trend has fallen out of favor recently.

8.4 COMPLETENESS

Completeness is one of the two candidate metrics that Luyten and Coninx [2005] propose for determining the usability of distributed interaction spaces, adding that it is impossible to ensure completeness when presenting content on devices with unpredictable variations in characteristics. Systems like ANMoLE and MIDAS relax the constraint of presenting specific content forms in order to provide additional certainty in ensuring completeness. Both these systems support presentation-time content modification to deliver the content that is most suitable for rendering on available devices. While ANMoLE relies on delivery-time modification alone, MIDAS takes a human-centered approach by involving authors as well as viewers in the content adaptation process.

The Resource Manager enables authors to visualize the device classes that could satisfactorily present available resource instances. This visualization aids authors in creating instances for presentation on device classes that may not render effectively the content associated with their hypertexts. When possible, the Resource Manager automatically or semi-automatically converts instances to support additional device classes. The authors have an opportunity to accept or reject the converted forms. Human validation of converted content is critical issue for maintaining the integrity of content from a human perspective. This two-pronged approach helps authors create content for effective presentation in varied contexts while lowering their effort by proposing automated content changes, thus lowering the authors' cognitive load as well as effort in converting content to various specifications. While it is impossible to predict the properties of devices that viewers will employ to access MIDAS hypertexts, the Resource Manager enables authors to approve acceptable content elements at authoring time in order to minimize the degree of presentation-time content adaptation. The Resource Manager also allows authors to specify a brief textual description for each resource id for displaying as a last resort, when none of the content representations are suitable for delivery to a user's set of devices.

Resource ids help decouple hypertext structure from the presentation of the information content. MIDAS guarantees that the content associated with each id will be presented, provided that the hypertext author has included the textual description. When adapting content during a browsing session MIDAS employs policies that have been informed by human perceptual abilities and preferences, especially for images. The policies for presentation of other media forms currently use media attributes alone. These policies could be further refined to include human perceptual properties.

8.5 COVERAGE

The Device Manager implements several policies for allocating content instances to devices. Different policies allow for different forms of coverage. As a policy can be chosen when the Device Manager initializes, each invocation of the application has the potential to support the coverage of devices to whatever extent the user desires.

The MIDAS optimal allocation policy takes a system-wide view, selecting content instances that make the best use of the resources available on all devices. Sometimes, it may sacrifice the best scoring instance of a resource for a particular device if selection of this instance lowers the total score for all selected instances. The selected instance may not only be a sub-optimal choice, it may be presented on a different device depending upon its score. This policy thus foregoes the best decisions for individual instances in favor of the best global matching of instances with devices.

The content optimal allocation policy takes the opposite approach; it always optimizes local coverage, at the risk of sacrificing system-wide optimization of coverage. It weighs each resource and directs the best scoring instance to the device that it matches. This policy may result in the starvation of less endowed devices like mobile phones with limited bandwidth and display space while all instances are routed to a powerful desktop computer that belongs to the MIDAS space. In this scenario, the user would interact with a single device, much as she does with typical Web browsing, with the option of switching to a different set of devices once it is no longer feasible for her to access the desktop computer. The other potential side effect of this policy is more concerning. When much of the content is routed to a single device, this device may be overloaded with content instances. While a particular instance may not tax the device, the task of presenting several instances may overwhelm an appliance, resulting in a breakdown of the MIDAS session.

The replication-based policies provide redundancy of information presentation at the cost of coverage. Rendering an instance of each resource on every available device, whether or not the policy accounts for interaction mode, is overkill and has the potential to overwhelm the devices with fewest resources. Furthermore, the human element involved in accessing multiple content forms associated with a resource remains to be explored. While multi-device interfaces have focused on presenting the content on multiple devices, none have explored simultaneous, alternate renditions of potentially interchangeable content. These policies provide an opportunity to explore the benefits and impediments of multi-device, multi-form presentation of related content.

Multibrowsing and the Ubiquitous Display System allow users to direct content to specific devices manually. The property of coverage is not applicable to these systems. Applications developed by the Pebbles project extend the display of a desktop computer or use a handheld for controlling presentations. In either of these cases there is little flexibility in the role played by each device. While caT allows clients from multiple devices to connect to the server the system does not conceptualize these in terms of devices. The server forwards the content to be displayed to all connected clients. MIDAS' replicated content policies mimic this behavior, while accounting for the various devices connected to the Device Manager. WebSplitter, ANMoLE, and IDD attempt to optimize the presentation of the content on the available devices. While WebSplitter and IDD provide coverage without altering content forms, ANMoLE and MIDAS support content reformulation.

8.6 CONVERSION

Web-based systems provide access to their contents by referring directly to instantiations in their final form, for example a particular jpeg file. In contrast, MIDAS introduces a degree of indirection through the use of resource ids, which the Device Manager resolves on demand to return the best available instance. Neither of these approaches guarantees that content can be presented as intended by its author on every device. MIDAS addresses this situation in two steps: the Resource Manager supports authors in converting content and the target devices display content within the device constraints, thus implicitly modifying the contents at presentation time.

Authors have an opportunity to vet the conversions generated by the Resource Manager and reject those instances that they deem unsatisfactory. The Resource Manager performs syntactic conversions. For example it scales down images and reduces color bits. It makes no determination regarding the appropriateness of the product of this conversion, leaving it to the author's judgment. When these approved instances must be further modified, the Device Manager's policies minimize the modifications by selecting instances that are the closest to the device characteristics.

The ANMoLE framework also envisions converting content by separating the audio and video streams during a video conferencing session to present partial content when such presentation is the best option. Other multi-device systems include single instances of final form content in their data specification and do not address the issue of conversion.

8.7 COMPOSITION

Infrastructures that build on Web technologies adopt a filtering-based approach, for example WebSplitter and IDD. Systems that build from the ground up adopt a generative approach, as these are not constrained by the characteristics of a system designed for a single device environment.

Building on the separation of structure, content, and presentation that is inherent in caT, MIDAS takes a generative approach to presenting content on multiple devices. The absence of a "master" presentation scheme that relates content within a page gives MIDAS great flexibility in routing content to the available devices. When appropriate, the individual components can be combined handily into commonly recognizable formats, as evidenced by the MIDAS Web interface. Within the constraints imposed by Web technologies, this interface functions as a fully compliant MIDAS browser.

MIDAS is designed to function as a middleware and does not require caT's conceptualization of hypertext structures. However, it subscribes to the principle of separation between structure and content. Adapting the MIDAS server for interaction with Web servers will likely result in an architecture where the Device Manager acts as a server component, much like a PHP engine or a content management system than a typical Web browser that receives prepackaged content for presentation to its viewers.

8.8 COHERENCE

MIDAS' generative approach to content presentation ensures that the relevant information content will be presented but does not attempt to organize this content in a particular layout. Its infrastructure and policies together ensure that the presentation is always consistent with the current state of the hypertext and the instances associated with all the active resources are presented. User actions, changes in the state of the environment (such as location and time), and availability or unavailability of devices are all situations that threaten the integrity of the presentation. When the state of the hypertext changes, MIDAS ensures that content associated with resources that are no longer active is removed from the presentation and that which is associated with the resources that are active in the new state is rendered.

When a user interacts with the hypertext, thus changing its state, she typically does so on one device. When a user action is received, the Device Manager directs all connected devices to suspend accepting further input as these devices no longer reflect the most recent state of the system. Simultaneously, it forwards the user action to the information service (caT server) and retrieves the new state of the hypertext. Once the instances associated with the new state are sent to the connected devices, the Device Manager instructs them to begin accepting user input again as these devices now reflect the user's current state of browsing. Maintaining a consistent view across all the connected devices is critical for ensuring that users always interact with the current state of the hypertext.

When a device is removed from the MIDAS space, a similar situation occurs. As the content presented on this device is now unavailable to the user, the information presentation can no longer be guaranteed to be coherent. The Device Manager reallocates the active content and routes the new content elements for presentation on the available devices. In this case also, the user input is suspended from the instant the Device Manager is notified of the device leaving and is enabled only when the Browser Coordinators receive the updated content elements.

Another factor that affects the coherence of information rendered is the device selection policy in use. When content optimal or MIDAS optimal policies are in force, the content associated with each active element is presented only once. However, when a replication-based policy is in effect, the connected devices each present the content elements in forms that are best suited for rendering on the target device. Different devices may present mutually conflicting content forms. MIDAS relies upon the content authors to address this issue. The authors are expected to group under a resource id only content forms that may be presented interchangeably.

8.9 COORDINATION

When the state of available content or of available devices changes, MIDAS responds by reapplying its policies to generate a stable presentation. It computes the scores for the new set of resource instances in the context of the new set of devices and reallocates the instances to devices using the current policy. This approach constitutes the most comprehensive coordination scheme among multi-device infrastructure. Building on caT's response scheme, which includes adapting the presentation to user characteristics and actions as well as to changes in location and time, MIDAS adds the ability to respond to changes in the device space.

Not all events jeopardize the coherence of the presentation equally. The availability of a new device without the corresponding unavailability of a device that is currently rendering content is one such situation. The new device could simply be ignored until the next significant change of state, when the active elements must be redistributed. However, this approach does not provide a discernable feedback to a user who just added a new device to the MIDAS space. Favoring a responsive approach over an efficient one, MIDAS reapplies its policies in reaction to all changes regardless of whether they currently constitute a critical change, one that the system must address immediately.

8.10 CONTINUITY

In addition to completeness, continuity is the other metric proposed by Luyten and Coninx [2005] for evaluating the usability of multi-device systems. Each of the variables that induce MIDAS to coordinate the presentation—user actions, changes to environmental conditions, and availability of devices—also has the potential to affect continuity. User initiated actions are unlikely to affect continuity seriously as a user may expect a response to her actions. However, system-initiated changes to the browsing state, such as a change in available actions during peak and off-peak hours, may be subtle and have the potential to surprise users who are not expecting such changes.

As their names imply, the content optimal and MIDAS optimal device selection policies favor optimization of available device properties by favoring the best matching instances. However, when a user adds a new device to a space or removes a current device, the reallocation of all elements is superfluous and potentially detrimental to preserving the continuity of the browsing session. Variations of the existing policies could support partial allocation of resources to devices without reallocating resources that are presented on devices whose status is unaffected by the change.

The replication-based policies do not suffer from this drawback, as the instances presented in the current state are likely to remain the best candidates in the new states as well, thus minimizing disruption to user activity. These policies, thus, employ instance redundancy as a mechanism to overcome a potential loss of continuity.

8.11 CONSTANCY

Infrastructures that tightly couple hypertext structure to information content support constancy by virtue of presenting the sole available instance regardless of the device context. The association of multiple instances with a resource in MIDAS or the on demand modification of content in ANMoLE hinders these systems in providing a consistent, familiar view of the content across different sessions or in scenarios where different devices are available during sessions.

Whether device change within a session or across two sessions, a user may have access to devices that are capable of rendering instances that she has seen before in order to help jog her memory or bring a sense of familiarity to a browsing session. MIDAS' current policies recalculate the scores for available instances and pair these with the available devices in order to find the most optimal presentation using the effective allocation policy. While the current policies focus on selecting content based on their conformance with device characteristics and human perception, these policies could incorporate support for constancy with minor modifications. An evaluation of MIDAS' usability as well as an exploration of the potential for multi-device access in real-life browsing and searching tasks will be incomplete without investigating the role of human memory in assessing the effectiveness and acceptance of multi-device presentations.

8.12 CONFIDENCE

MIDAS relies upon the reliability underlying networking infrastructure for maintaining the integrity of the communication. The use of connection-oriented sockets minimizes the chances of messages being lost or garbled. The communication agents can recover from such losses as MIDAS incorporates messages for requesting components to resend lost messages. Beyond such recovery, MIDAS does not address the issues of trust and security. All information is transmitted using a Web-safe character encoding and may easily be intercepted by third parties.

MIDAS is designed to be a flexible, configurable multi-device system. The Device Manager and the Browser Coordinators are inter-communicating, independently configurable units. While the Device Manager sets policies that affect the overall presentation and interaction within the system, the Browser Coordinators configure their host devices to play only certain roles. Users may directly affect MIDAS' behavior by expressing their intent for using each device by setting the appropriate Browser Coordinator properties. MIDAS presentations respond to user actions, changes in a user's location and system time, as well as to the characteristics of the available devices. MIDAS thus serves as a substrate for exploring various attributes of multi-device spaces, articulated by the 12C framework.

The following chapter summarizes the contributions of this research to the area of multi-device hypertext systems and outlines directions for continuing this work.

CHAPTER IX

CONCLUSION

In this dissertation, I have presented the 12C framework and the MIDAS system. The framework consists of twelve criteria for describing and comparing multi-device interactive systems—systems that distribute the presentation of information content over several devices simultaneously and, possibly, allow users to interact with this presentation from several devices as well. MIDAS is an implementation of a flexible, multi-device system that enables the specification of a variety of behaviors for presenting content as well as for supporting user interaction.

9.1 CONTRIBUTIONS

My research advances the state-of-the-art in multi-device user interfaces by contributing to the advancement of theory, system infrastructure, and presentation techniques. Aspects of this work inform the areas of Hypertext Systems, Pervasive Computing, Human-Centered Computing, and Digital Libraries—areas that focus on presenting information for human use in particular contexts.

9.1.1 Theory

The 12C framework provides criteria for grounding the discussion of various aspects of multi-device systems. An earlier attempt focused on defining criteria to assess their usability alone [Luyten and Coninx 2005]. The IDD framework defined completeness and continuity as the two relevant parameters for evaluating multi-device system usability. The 12C framework extends the criteria that describe usability to include two other attributes: Coherence and Constancy. Taking a wider view, the 12C framework includes attributes for describing additional properties of multi-device system properties, such as infrastructure (Comity, Completeness), presentation characteristics (Concurrency, Coverage, Conversion, Coherence, Coordination, Continuity), interaction (Control, Continuity), interface (Coverage, Composition), and security (Confidence).

The user study I conducted provided an insight into the human perception of similarity when image characteristics vary. The study demonstrated that while presenting modified images, the nature of content lost (color depth vs. space) impacts users' views regarding the suitability of presenting lossy images. While quantitatively more content is lost when scaling images, users tend to prefer scaling to color reduction. Furthermore, not all images scale equally well as image viewers are interested in particular aspects of images, for example human faces and textual elements. Fortunately, automated techniques for identifying these features are readily available. Thus, a system could scan images to test for the inclusion of these features, label them accordingly, and based on feature-specific policies, could adapt images appropriately—all without direct attention from a human moderator. Another key insight from the study is that color images, when presented in grayscale, were perceived to be closest to the originals when the color loss was not coupled with scaling or bit-depth reduction.

9.1.2 Techniques

I have applied the nearness scores provided by the study participants to develop metrics for use in image presentation techniques. These metrics adapt images for presentation on various devices based on both the device characteristics and human perceptual properties. The image transformation technique applies experimentally determined weights to three transcoding mechanisms: scaling; color reduction; and gray scaling. The scoring formula factors in image type when scaling in order to reflect the type-specific differences identified by the study participants.

The typical display resolutions for desktop computers and notebook computers have evolved. These appliances now come equipped with widescreen displays that have different aspect ratios. However, new classes of smaller devices now use the display resolutions included in the study. Apple's iPad and the HP TouchPad ship with 1024 X 768 pixel displays and several mid-range smart phone displays consist of 320 X 480 pixels. While the video standard has shifted to HD digital cameras typically click pictures in the 4:3 aspect ratio, which was used for images in this study. While the specifications of the display and image capture technologies will continue to change, the key contribution of this research is the principle of using human perception-based metrics as a factor in content adaptation.

9.1.3 System

The implemented infrastructure, MIDAS, is capable of distributing content and interaction elements over a set of devices with heterogeneous characteristics. While the distribution of elements over multiple devices is not unique in itself, MIDAS serves as a flexible infrastructure, lending itself to the exploration of several content distribution and interaction strategies. The MIDAS architecture supports independent configuration of server-side (Device Manager) and client-side (Browser Coordinator) elements to facilitate this flexibility.

MIDAS instantiates the human perception-based image adaptation technique, making it the first multi-device system to include human perception as a factor in its content adaptation strategy. MIDAS is also the first system that implements content redistribution when devices become available or unavailable. While several prior projects have identified the necessity to support content reorganization in response to device space changes [Han et al. 2000; Haneef and Ganz 2004; Luyten and Coninx 2005] none have reported on a working implementation. In addition to changes in available devices, MIDAS' integration with caT enables it to reallocate content in response to stimuli from the users as well as environmental properties such as location and time.

In placing a minimal overhead for inclusion in the MIDAS space, the architecture is designed to incorporate appliances with diverse properties. Each appliance must run a properly configured instance of the Browser Coordinator in order to be included in the device space. MIDAS further encourages diversity in device characteristics by expressing content adaptation and allocation policies in terms of the differences between characteristics of content instances and those of the available devices. This approach focuses on rendering the best possible instance regardless of a device's properties and is inherently extensible to include new devices without a necessity to codify new policies for new devices.

9.2 FUTURE WORK

The current MIDAS prototype demonstrates a proof-of-concept system that addresses my research questions and embodies the six design goals listed in Chapter I. Having developed a prototype, I am eager to exploit its potential in exploring additional questions that follow from this initial investigation. I anticipate continuing work on this research in exploring the nature and form of media, the effects of content transformation on human perception, and enhancement of policies for supporting multi-device interaction. The following subsections outline particular directions for continuing this work.

9.2.1 Infrastructure

The current MIDAS implementation uses caT's traditional browsers to demonstrate its ability to act as a middleware, supporting its seamless insertion within caT's client-server architecture. MIDAS' Web interface, on the other hand, demonstrates that it can span architectures, enabling caT hypertext access from commonly used Web browsers. While historically, Web servers have not supported a reliable push mechanism that would enable a server to proactively transmit content to client devices, emerging technologies such as the WebSockets API [Hickson 2011] are quickly maturing, enabling Web servers to push content to clients. This architectural improvement makes Web servers an attractive information source due to the large audience that Web enjoys.

Google's Android and Apple's iOS have emerged as the *de facto* mobile application development standards for smart phones as well as tablet computers and e-book readers. Android is popular due to its broad adoption by several hardware manufacturers, amounting to a 40% share of the U.S. smartphone market while Apple's iOS holds 28% [Lawson 2011]. Both platforms enable users to download and install custom applications. Developing Browser Coordinators for these two platforms will enable MIDAS access to a large user population. Both Android and iOS power smartphones as well as tablet computers, thus, the development of Browser Coordinators will enable users to connect to a Device Manager from a diverse devices.

MIDAS' vision to support the inclusion of public devices requires a feature that enables a user to request a remote device to join her space. Google TV [Google 2011], with its ability to run Android application, fits the bill as a widely adopted platform that MIDAS could employ for remote invocation of services. Inclusion of public devices, similar to that demonstrated by the Ubiquitous Display Project [Aizawa et al. 2002], will open the gates to exploring issues in the use of channeling content to shared devices as well as the host of social and privacy implications that such a presentation involves.

9.2.2 Interaction and usability issues

The infrastructural improvements proposed above will greatly enhance the prospects for evaluation of MIDAS in a realistic, multi-device use cases. A usability evaluation, especially one that includes users interacting over contemporary devices will highlight issues of user attention and cognitive overheads involved in switching between and collating information presented across multiple devices. Taking a human-centered approach, I intend to evaluate features that target particular research questions through controlled studies in laboratory settings as well as through long-term, monitored usage "in the wild".

Systems like WebSplitter [Han et al. 2000] have reported multi-device presentations in collaborative settings with one presenter and several viewers. I would like to study the use of shared device spaces by combing the devices available to users in mobile settings such as addressing the information needs of paleontologists while they are searching for fossils in the field. This setting will enable a small group of users to combine devices for satisfying shared information in support of shared goal.

9.2.3 Content transformation

In the pilot study that I conducted, participants indicated a preference for some kinds of image transformations to others, for example scaling was preferable to color reduction. While this study focused only on scaling as a means of reducing the spatial area of images, a variety of transformations—and combination of such transformations—are possible. Some of these approaches include cropping before scaling [Suh et al. 2003], a visual attention-based model for image cropping [Chen et al. 2003], and automatic image retargeting [Setlur et al. 2005]. Transformations that are specifically crafted for devices with particular characteristics may fare better for adapting content to devices than the general-purpose approaches that my study focused investigated.

While the current scoring policies for other content forms such as text, audio, and video are primarily based on document properties, conferences such as ACM ASSETS [ASSETS 2011] have explored issues in the transformation of content with a view to conveying their essence to disabled populations. Lessons learned from adapting content for communities with special needs could provide an insight on content features that must be preserved or accounted for when modifying instances for presentation on diverse devices. Transformation of content across forms is also an attractive approach for multi-device presentations. For example, text files could be rendered as audio streams using software such as Festival [Festival 2011] for automatic text-to-speech conversion when a user only has access to mobile devices with small screens. Ma et al.'s [2000] framework for adaptation of content will serve as a valuable guide for systematically exploring the variety of possible content transformations.

Transformation of content across forms raises situations where content scoring algorithms must compare scores generated using form-specific formulae. For example, how does an image instance score of 0.3 compare against a text instance score of 0.3? How would a system normalize cross-content scores?

Cross-form content transformation also has implications for device selection policies. Would it be advantageous to present an instance in its original form or transform it to another form and improve the overall presentation score while lowering that for the current instance? Questions such as these will result in more involved device selection policies. Furthermore, on-the-fly conversion results in a delay when delivering content to users. How much performance hit would a user tolerate? Responses to such questions will probably be situation-specific and only gained through experimentation.

Technology becomes usable only when it vanishes into the background [Weiser 1991]. My ultimate goal is for MIDAS to act as a distributed interface that enables the devices to vanish into the background and for users to interact with the content seamlessly regardless of the diversity device characteristics.

REFERENCES

AIZAWA, K., KENTARO, K., AND NAKAHIRA, K. 2002. Ubiquitous displays for cellular phone-based personal information environments. In *Proceedings of the Third IEEE Pacific Rim Conference on Multimedia (PCM 2002).* Lecture Notes in Computer Science, vol. 2532. Springer-Verlag, 25–32.

AMAZON. 2011. Kindle 3G wireless reading device. *http://www.amazon.com/Kindle-Wireless-Reader-3G-Wifi-Graphite/dp/B002FQJT3Q*. Accessed July 2011.

ANDERSON, C., DOMINGOS, P., AND WELD, D. 2001. Personalizing web sites for mobile users. In *Proceedings of the 10th International World Wide Web Conference (WWW '01).* ACM Press, 565–575.

ASSETS. 2011. ACM SIGACCESS conference on computers and accessibility. *http://dl.acm.org/event.cfm?id=RE205&CFID=40549126&CFTOKEN=42370432*. Accessed July 2011.

BANDELLONI, R. AND PATERNÒ, F. 2004. Flexible interface migration. In *Proceedings of the Ninth International Conference on Intelligent User Interfaces (IUI'04)*. ACM Press, 148–155.

BARNES & NOBLE. 2011. All-new NOOK tech. specs. *http://www.barnesandnoble.com/nook/features/techspecs/index.asp*. Accessed July 2011.

BERTI, S., CORREANI, F., MORI, G., PATERNÒ, F., AND SANTORO, C. 2004. TERESA: a transformation-based environment for designing and developing multi-device interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*. ACM Press, 793–794.

BICKMORE, T. AND SCHILIT, B. 1997. Digestor: device-independent access to the world-wide web. In *Proceedings of the Sixth International World Wide Web Conference (WWW6) http://www.ra.ethz.ch/cdstore/www6/Technical/Paper177/Paper177.html*. Accessed June 2011.

BOS, B., ÇELİK, T., HICKSON, I., AND LIE, H. (ED.S). 2011. Cascading style sheets, Level 2 Revision 1 (CSS 2.1) specification. *http://www.w3.org/TR/CSS21/*. Accessed June 2011.

BULTERMAN, D., JANSEN, J., CESAR, P., MULLENDER, S., HYCHE, E., DEMEGLIO, M., QUINT, J., KAWAMURA, H., WECK, D., GARCÍA PAÑEDA, X., MELENDI, D., CRUZ-LARA, S., HANCLIK, M., ZUCKER, D., AND MICHEL, T. 2008. Synchronized multimedia integration language (SMIL 3.0). *http://www.w3.org/TR/SMIL/*. Accessed June 2011.

BUYUKKOKTEN, O., KALJUVEE, O., GARCIA-MOLINA, H., PAEPCKE, A., AND WINOGRAD, T. 2002. Efficient web browsing on handheld devices using page and form summarization. *ACM Trans. Inform. Syst. 20,* 1, 82–115.

CAI, D., YU, S., WEN, J.-R., AND MA, W.-Y. 2003. Extracting content structure for web pages based on visual representation. In *Proceedings of the 5th Asia Pacific Web Conference (APWeb '03)*. Springer-Verlag, 406–417.

CHEN, L., XIE, X., FAN, X., MA, W., ZHANG, H., AND ZHOU H. 2003. A visual attention model for adapting images on small displays. *Multimedia Syst. J. 9,* 4, 353–364.

CHUA, H., SCOTT, S., AND CHOI, Y. 2007. Enabling device independent co-browsing through adaptive view points. *J. of Computer and Syst. Sciences 74,* 2, 146–157.

CLARK, J. 1999. XSL transformations (XSLT) Version 1.0. *http://www.w3.org/TR/xslt.* Accessed June 2011.

CUI, Y., NAHRSTEDT, K., AND XU, D. 2004. Seamless user-level handoff in ubiquitous multimedia service delivery. *J. of Multimedia Tools and App.s 22,* 2, 137–170.

DÖRING, T., SHIRAZI A., AND SCHMIDT, A. 2010. Exploring gesture-based interaction techniques in multi-display environments with mobile phones and a multi-touch table. In *Proceedings of the Workshop on Coupled Display Visual Interfaces (PPD10).* HIT Lab, 47–50.

DSPACE. 2011. Functional overview- DSpace documentation – DuraSpace wiki. *https://wiki.duraspace.org/display/DSDOC/Functional+Overview.* Accessed June 2011.

ECKEL, B. 1993. *C++ Inside & Out*, Osborne McGraw- Hill, Berkeley, CA.

FESTIVAL. 2011. The Festival speech synthesis system. *http://www.cstr.ed.ac.uk/projects/festival/*. Accessed July 2011.

FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. 1999. Hypertext transfer protocol. *http://www.w3.org/Protocols/rfc2616/rfc2616.txt*. Accessed June 2011.

FLICKR. 2011. Flickr: help: photos. *http://www.flickr.com/help/photos/*. Accessed July 2011.

FOX, A. AND BREWER, E. 1996. Reducing WWW latency and bandwidth requirements by real-time distillation. In *Proceedings of the Fifth International World Wide Web Conference (WWW5).* Elsevier Science Publishers, 1445–1456.

FREDERICK, G. AND LAL, R. 2010. Device awareness and content adaptation. *Beginning Smartphone Web Development,* Springer-Verlag, New York, NY, 97–134.

FREEMAN, E. AND FREEMAN, E. 2006. *Head First HTML with CSS and XHTML*. O'Reilly Media, Inc. Sebastopol, CA.

FURUTA, R. 2005. Separation of concerns in hypertext: articulation points that increase flexibility. *From Integrated Publication and Information Systems to Information and Knowledge Environments.* Lecture Notes in Computer Science, vol. 3379, Hemmje, M., Risse, T., and Niederée, C. (Ed.s). Springer-Verlag, 128–137.

FURUTA, R. AND NA, J.-C. 2002. Applying caT's programmable browsing semantics to specify world-wide web documents that reflect place, time, reader, and community. In *Proceedings of the 2002 ACM Symposium on Document Engineering (DocEng '02)*. ACM Press, 10–17.

FURUTA, R. AND STOTTS, P. 1994. Interpreted collaboration protocols and their use in groupware prototyping. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*. ACM Press, 121–132.

GANTENBEIN, R. AND JONES, D. 1986. Dynamic binding of separately compiled objects under program control. In *Proceedings of the 1986 ACM Fourteenth Annual Conference on Computer Science (CSC '86)*. ACM Press, 287–292.

GHIANI, G., PATERNÒ, F., AND SANTORO, C. 2010. On-demand cross-device interface components migration. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '10).* ACM Press, 299–308.

GILBERT, B., GOODE, A., AND SATYANARAYANAN, M. 2010. Pocket ISR: virtual machines anywhere. Report CMU-CS-10-112, Carnegie Mellon University School of Computer Science, Pittsburgh, PA.

GIRGENSOHN, A. AND LEE, A. 2002. Making web sites be places for social interaction. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*. ACM Press, 136–145.

GONÇALVES, M., FOX, E., WATSON, L., AND KIPP, N. 2004. Streams, structures, spaces, scenarios, societies (5S): A Formal Model for Digital Libraries. *ACM Trans. Inform. Syst. 22,* 2, 270–312.

GOOGLE. 2011. Features – Google TV. *http://www.google.com/tv/features.html*. Accessed July 2011.

GREEN, C. 2010. Senior design project: iPad multimodal rummy. *http://www.codygreen.com/2010/04/senior-design-project-ipad-multimodal-rummy/*. Accessed February 2011.

GROLAUX, D., VAN ROY, P., AND VANDERDONCKT, J. 2004. Migratable user interfaces: beyond migratory interfaces. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems Networking and Services (MobiQuitous 2004)*. IEEE Press, 422–300.

GUIRGUIS, S. AND HASSAN, M. 2010. A smart framework for web content and resources adaptation in mobile devices. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10)*. IEEE Press, 487–492.

GUPTA, S., KAISER, G., NEISTADT, D., AND GRIMM, P. 2003. DOM-based content extraction of HTML documents. In *Proceedings of the 12th International World Wide Web Conference (WWW '03)*. ACM Press, 207–214.

HALASZ, F. AND SCHWARTZ, M. 1994. The Dexter hypertext reference model. *Comm. ACM 37,* 2, 30–39.

HAN, R., PERRET, V., AND NAGHSHINEH, M. 2000. WebSplitter: a unified XML framework for multi-device collaborative web browsing. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW '00)*. ACM Press, 221–230.

HANEEF, A. AND GANZ, A. 2004. ANMoLe - An adaptive multimedia content delivery middleware architecture for heterogeneous mobile multi-device neighborhoods. *Intern. J. Multimedia Tools and App.s 22*, 2, 171–186.

HICKSON, I. 2011. The WebSocket API. *http://dev.w3.org/html5/websockets/*. Accessed July 2011.

HJELMÅS, E. AND LOW, B. 2001. Face detection: a survey. *Comp. Vision Image Understanding 83,* 3, 236–274.

HOGAN, B. 2010. *HTML5 & CSS3: Develop with Tomorrow's Standards Today*. Pragmatic Bookshelf, Raleigh, NC.

HOH, S., GILLIES, S., AND GARDNER, M. 2003. Device personalisation — where content meets device. *BT Tech. J. 21,* 1, 67–74.

VAN DEN HOVEN, E. AND EGGEN, B. 2004. Tangible computing in everyday life: extending current frameworks for tangible user interfaces with personal objects. In *Proceedings of Ambient Intelligence: Second European Symposium (EUSAI 2004).* Lecture Notes in Computer Science, vol. 3295. Springer-Verlag, 230–242.

HOWARD, W. 2004. Better displays with organic films. *Sci. Am. 209,* 2, 76–81.

INFOWORLD. 2011. 4G defined: WiMax and LTE don't qualify. *http://www.infoworld.com/d/mobilize/4g-defined-wimax-and-lte-dont-qualify-683*. Accessed January 2011.

JANOUSEK, S., ELROM, E., AND JOOS, T. 2009. *AdvancED Flash on Devices: Mobile Development with Flash Lite and Flash 10,* Springer-Verlag, New York, NY.

JENSEN, K. 1992. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use Volume 1*. Springer-Verlag, New York, NY.

JOHANSEN, R. 1988. *Groupware: Computer Support for Business Teams*. Free Press, New York, NY.

JOHANSON, B., PONNEKANTI, S., SENGUPTA, C., AND FOX, A. 2001. Multibrowsing: moving web content across multiple displays. In *Proceedings of Ubiquitous Computing Conference (UbiComp 2001).* Lecture Notes in Computer Science, vol. 2201. Springer-Verlag, 346–353.

JOHNSON, L., SMITH, R., LEVINE, A., AND HAYWOOD, K. 2010. Four to five years: flexible displays. *The 2010 Horizon Report: K-12 Edition.* The New Media Consortium, Austin, TX.

KARADKAR, U., FURUTA, R., USTUN, S., PARK, Y.-J., NA, J.-C., GUPTA, V., CIFTCI, T., AND PARK, Y. 2004. Display-agnostic hypermedia. In *Proceedings of the Fifteenth ACM Conference on Hypertext and Hypermedia (Hypertext '04).* ACM Press, 233–242.

KARADKAR, U., FURUTA, R., JOHN, J., AND NA, J.-C. 2005. Exploring user perceptions of digital image similarity. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05).* ACM Press, 89–90.

KIM, K., JUNG, K., AND KIM, J. 2003. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Trans. Pattern Analysis and Machine Intel. 25,* 12, 1631–1639.

KLOMPMAKER, F., SCHRAGE, K., AND REIMANN, C. 2009. INDiE: a framework for human computer interaction in distributed environments. In *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems (Mobility '09)*. ACM Press, Article No. 26.

KLYNE, G., REYNOLDS, F., WOODROW, C., OHTO, H., HJELM, J., BUTLER, M., AND TRAN, L. 2010. Composite capability/preference profiles (CC/PP): structure and vocabularies 1.0. *http://www.w3.org/TR/CCPP-struct-vocab/*. Accessed June 2011.

KOZUCH, M. AND SATYANARAYANAN, M. 2002. Internet suspend/resume. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02).* IEEE Press, 40–46.

KRISTOL, D. AND MONTULLI, L. 2000. HTTP state management mechanism. *http://www.ietf.org/rfc/rfc2965.txt*. Accessed July 2011.

LAWSON, S. 2011. Nielson: Android grows, RIM draws attention. *http://www.pcworld.com/article/239394/nielsen_android_grows_rim_draws_attention.html*. Accessed June 2011.

LIE, H., ÇELIK, T., GLAZMAN, D., AND VAN KESTEREN, A. (ED.S). 2011. Media queries. *http://www.w3.org/TR/css3-mediaqueries/*. Accessed January 2011.

LORETO, S., SAINT-ANDRE, P., SALSANO, S., AND WILKINS, G. 2011. Known issues and best practices for the use of long polling and streaming in bidirectional HTTP. *http://tools.ietf.org/html/rfc6202*. Accessed July 2011.

LUYTEN, K. AND CONINX, K. 2005. Distributed user interface elements to support smart interaction spaces. In *Proceedings of the Seventh IEEE International Symposium on Multimedia (ISM '05).* IEEE Computer Society, 277–286.

Ma, W., Bedner, I., Chang, G., Kuchinsky, A., and Zhang, H. 2000. A framework for adaptive content delivery in heterogeneous network environments. In *Proceedings of SPIE Multimedia Computing and Networking*. SPIE, 86–100.

MANCA, M. AND PATERNÒ, F. 2011. Distributing user interfaces with MARIA. In *Proceedings of Distributed User Interfaces 2011 (DUI 2011),* CHI 2011 Workshop, *http://giove.isti.cnr.it/attachments/publications/chi11%20workshop%20distributed%20ui.pdf*. Accessed July 2011.

MESKENS, J., LUYTEN, K., AND CONINX, K. 2010. DMacs: building multidevice user interfaces by demonstrating, sharing and replaying design actions. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM Press, 129–138.

MICROSOFT. 2008. Sheraton hotels and resorts transforms the hotel lobby experience with Microsoft Surface. *http://www.microsoft.com/Presspass/press/2008/aug08/08-13SheratonMSSurfacePR.mspx*. Accessed June 2011.

MICROSOFT. 2011a. Welcome to Surface. *http://www.microsoft.com/surface/.* Accessed June 2011.

MICROSOFT. 2011b. Windows Phone 7. *http://www.microsoft.com/windowsphone/en-us/default.aspx*. Accessed July 2011.

MILLER, R. AND MYERS, B. 1999. Synchronizing clipboards of multiple computers. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology (UIST'99).* ACM Press, 65–66.

MOGENSEN, P., KOIVISTO T., PEDERSEN, K., KOVÁCS, I., RAAF, B., PAJUKOSKI, K., AND RINNE, M. 2009. LTE-advanced: the path towards gigabit/s in wireless mobile communications. In *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE).* IEEE Press, 147–151.

MOMA. 2011. MoMA – The collection – Sarah Morris – creative artists legacy. *http://www.moma.org*. Accessed June 2011.

MORI, G., PATERNÒ, F., AND SANTORO, C. 2003. Tool support for designing nomadic applications. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI '03)*. ACM Press, 141–148.

MYERS, B. 2001. Using handhelds and PCs together. *Comm. ACM 44,* 11, 34–41.

MYERS, B., LIE, K., AND YANG, B. 2000a. Two-handed input using a PDA and a mouse. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM Press, 41–48.

MYERS, B., MILLER, R., BOSTWICK, B., AND EVANKOVICH, C. 2000b. Extending the Windows desktop interface with connected handheld computers. In *Proceedings of the 4th USENIX Windows Systems Symposium*. Usenix, 79–88.

MYERS, B., STIEL, H., AND GARGIULO. R. 1998. Collaboration using multiple PDAs connected to a PC. In *Proceedings of the 1998 ACM Conference on Computer-Supported Cooperative Work (CSCW '98).* ACM Press, 285–294.

NA, J.-C. 2001. Context-aware hypermedia in a dynamically changing environment, supported by a high-level Petri net. Dissertation, Texas A&M University, College Station, TX.

OASIS. 2011. OASIS user interface markup language (UIML) TC. *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uiml*. Accessed June 2011.

ODIO, S. 2010. More beautiful photos. *http://blog.facebook.com/blog.php?post=432670242130*. Accessed July 2011.

PARK, Y. 2004. Resource Manager. Report, Dept. of Computer Science, Texas A&M University, College Station, TX.

PARK, Y. 2010. Augmenting users' task performance through workspace narrative exploration. Dissertation, Texas A&M University, College Station, TX.

PARK, Y., KARADKAR, U., AND FURUTA, R. 2010. Component-based authoring of complex, Petri net-based digital library infrastructure. In *Proceedings of the Fourteenth European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010).* Lecture Notes in Computer Science, vol. 6273. Springer-Verlag, 22–29.

PATERNÒ, F. AND SANTORO, C. 2003. A unified method for designing interactive systems adaptable to mobile and stationary platforms. *Interacting With Computers 15,* 3, 349–366.

PATERNÒ, F., SANTORO, C., AND SPANO, L. 2009. MARIA: a universal language for service-oriented applications in ubiquitous environment. *ACM Trans. Comp.-Human Interaction 16,* 4, 19:1–19:30.

PATERNÒ, F. AND ZICHITTELLA, G. 2010. Desktop-to-mobile web adaptation through customizable two-dimensional semantic redesign. In *Proceedings of the 3rd Conference on Human-Centred Software Engineering (HCSE 2010).* Lecture Notes in Computer Science, vol. 6409. Springer-Verlag, 79–94.

PENICHET, V., MARIN, I., GALLUD, J., LOZANO, M., AND TESORIERO, R. 2007. A classification method for CSCW systems. *Electronic Notes in Theoretical Computer Science 168*, 237–247.

PEREZ, S. 2008. Microsoft Surface now in AT&T stores. *http://channel9.msdn.com/blogs/coolstuff/microsoft-surface-now-in-atampt-stores*. Accessed June 2011.

PETERSON, J. 1981. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, Englewood Cliffs, N.J.

PHELPS, T. AND WILENSKY, R. 1996. Toward active, extensive, networked documents: multivalent architecture and applications. In *Proceedings of the First ACM International Conference on Digital Libraries (DL '96).* ACM Press, 100–108.

PHELPS, T. AND WILENSKY, R. 2001. The multivalent browser: a platform for new ideas. In *Proceedings of the 2001 ACM Symposium on Document Engineering (DocEng '01).* ACM Press, 58–67.

PQ LABS. 2011. Multi-touch screen for LCD large screen, plasma TV display - touch screen overlay. *http://multi-touch-screen.com/product.html*. Accessed June 2011.

RAGGETT, D., LE HORS, A., AND JACOBS, I. 1999. HTML 4.01 Specification. *http://www.w3.org/TR/html401/*. Accessed July 2011.

SÁNCHEZ, G., MENDOZA, S., DECOUCHANT, D., GALLARDO-LÓPEZ, L., AND RODRÍGUEZ, J. 2010. Plasticity of interaction interfaces: the study case of a collaborative whiteboard. In *Proceedings of the 16th CRIWG Conference on Collaboration and Technology (CRIWG 2010).* Lecture Notes in Computer Science, vol. 6257. Springer-Verlag, 265–280.

SEEBACH, P. 2008. The stateless state. http://www.ibm.com/developerworks/web/library/wa-state/. Accessed July 2011.

SETLUR, V., TAKAGI, S., RASKAR, R., GLEICHER, M., AND GOOCH, B. 2005. Automatic image retargeting. In *Proceedings of the 4th International Conference on Mobile and Ubiquitous Multimedia (MUM '05)*. ACM Press, 59–68.

SMALDONE, S., GILBERT, B., BILA, N., IFTODE, L., DE LARA, E., AND SATYANARAYANAN, M. 2009. Leveraging smart phones to reduce mobility footprints. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*. ACM Press, 109–122.

SMITH, K. 2010. Device independent authoring language (DIAL). *http://www.w3.org/TR/dial/*. Accessed June 2011.

SMITH, K. AND SANDERS, D. 2007. Device description repository requirements 1.0. *http://www.w3.org/TR/DDR-requirements/*. Accessed July 2011.

SONG, H., CHU, H., AND KURAKAKE, S. 2002. Browser session preservation and migration. *Poster Session of WWW 2002*. Hawaii, USA. 2.

SPRINT. 2011. 4G wireless broadband network from Sprint. *http://shop.sprint.com/en/solutions/mobile_broadband/mobile_broadband_4G.shtml*. Accessed January 2011.

STOTTS, P. AND FURUTA, R. 1989. Petri-net-based hypertext: document structure with browsing semantics. *ACM Trans. Inform. Syst. 7,* 1, 3–29.

SUH, B., LING, H., BEDERSON, B., AND JACOBS, D. 2003. Automatic thumbnail cropping and its effectiveness. In P*roceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03).* ACM Press, 95–104.

UAPROF. 2001. WAG UAProf. *http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-248-uaprof-20011020-a.pdf*. Accessed June 2011.

ULLMER, B. AND ISHII, H. 2000. Emerging frameworks for tangible user interfaces. *IBM Systems Journal 39,* 3.4, 915–931.

USTUN, S. 2003. Audio browsing of automaton-based hypertext. Masters Thesis, Texas A&M University, College Station, TX.

VERIZON. 2011. Verizon 4G. *http://waptag.net/ads/sites/V/verizon_4g_f/index.php.* Accessed January 2011.

VNC. 2011. RealVNC - VNC - the original cross-platform remote control solution. *http://www.realvnc.com/vnc/index.html*. Accessed July 2011.

W3SCHOOLS. 2011. HTTP meta http-equiv attribute. *http://www.w3schools.com/tags/att_meta_http_equiv.asp*. Accessed July 2011.

WEISER, M. 1991. The computer for the 21^{st} century. *Sci. Am. 265,* 3, 94–104.

WU, V., MANMATHA, R., AND RISEMAN, E. 1999. Textfinder: an automatic system to detect and recognize text in images. *IEEE Trans. Pattern Analysis and Machine Intel. 21,* 11, 1224–1229.

WURFL. 2011. WURFL by ScientiaMobile. *http://wurfl.sourceforge.net/*. Accessed June 2011.

X. 2011. X.Org wiki – home. *http://www.x.org/*. Accessed July 2011.

YE, Q., HUANG, Q., GAO, W., AND ZHAO, D. 2005. Fast and robust text detection in images and video frames. *Image and Vision Computing 23,* 565–576.

ZURAWSKI, R. AND ZHOU, M. 1994. Petri nets and industrial applications: a tutorial. *IEEE Trans. Industrial Electronics 41,* 6, 567–583.

APPENDIX A

FREEFORM CLASSIFICATION OF 100 DIGITAL PHOTOGRAPHS

In the first stage of the experiment, five test subjects were given 100 prints of digital photographs and asked to devise a classification scheme for these pictures. Table XII lists the categories that the subjects created for this task. The labels for the categories are reproduced verbatim—including the spelling, spacing, and capitalization as the subjects used it. The numbers in parentheses indicate the count of photographs the subject assigned to the category. These categories formed the basis for the four-way classification used in stage 2. The number of categories created is shown in square brackets adjacent to subject identifiers. The numbers in parentheses indicate the pictures assigned by the subject to that category.

Table XII. Freeform classification of 100 digital photographs

mie00011 [22]	mie00031 [40]	mie00026 [21]	mie00082 [12]	mie00059 [22]
Animals – artistic (2)	apartments (1)	adventure/camping holiday photos (5)	architecture (15)	animals (4)
Animals – general (3)	appliances/kitchen (3)	airshow and flight (4)	fantasy (1)	architectural detail-sights (4)
Architecture & objects (7)	bad pictures (7)	artistic (4)	fiesta (5)	architecture-beautiful (9)
children (9)	big city (2)	delete (1)	hmmm… (6)	architecture-not beautiful (2)
city/crowds –artistic (4)	Birds on Water (2)	domestic holidays (6)	household (3)	everyday items-beautiful (6)
city/crowds –general (4)	car/parking (4)	festive events (6)	nature (22)	everyday items-not beautiful (5)
Flowers – artistic (4)	children (10)	flower and nature (5)	nature and humans (11)	flowers-detail (3)
Flowers – general (2)	chruch/wedding (3)	friend's wedding (2)	people (21)	flowers-pattern/no detail (1)
Focus on architectural elements - interesting angles (5)	deer (2)	international trips/holidays (8)	religious (3)	no people-scenery-still beautiful (5)
Food (3)	Disneyland (1)	other (8)	sky (4)	no people-scenery-very beautiful (6)
land-based vehicles (5)	dragon (1)	personal household photos (9)	sports (3)	non-edible food-not beautiful (2)
landscape with human elements – artistic (2)	dude (2)	photos from one international trip (4)	Technology (6)	people-children (5)
landscape with human elements – general (7)	dudes (1)	relatives (8)		people-group-aware of picture being taken (6)
landscape without manmade – artistic (3)	east asian country? (2)	scenic (5)		people-group-not aware of picture being taken (7)
landscape without manmade – general (3)	flower (5)	seattle trip (2)		people-not further specified (4)
Misc. (6)	food (4)	sport (2)		people-single person-background matters (8)
Misc. household objects (4)	gardens/parks (4)	texas (4)		people-wedding (1)
people posing with some element (7)	India? (4)	trip to the show (2)		pictures of pictures - no third dimension (7)
people-general (10)	lake (1)	university days (8)		street scenes (5)

Table XII. continued

mie00011 [22]	mie00031 [40]	mie00026 [21]	mie00082 [19]	mie00059 [22]
pics of pics (4)	London (1)	visiting a friend's place (4)		surreal (2)
texture (2)	mountain (5)	wildlife (3)		taken from above ground-no details (1)
Why? (4)	New Orleans (4)			view (7)
	old architecture (2)			
	outdoor activities (2)			
	plane + shuttle (2)			
	running guys (1)			
	same guy (2)			
	San Francisco (1)			
	sculpture (1)			
	seattle (2)			
	signs (2)			
	slug (1)			
	sports cover/legend (1)			
	sunsets (3)			
	swimming pool (1)			
	Texas A&M (2)			
	town + mountain (1)			
	view from plane (2)			
	water + houses by water (3)			
	yellow and green (2)			

APPENDIX B

QUESTIONNAIRES

Test subjects completed the questionnaires shown in this appendix. The participants in stage 1 completed only the demographic questionnaire. Those in stage 2 completed all three questionnaires online. The questionnaires have been reformatted to fit within the margins of this document.

Information Element Relationship Evaluation

Pre-Task Questionnaire

Alias Used: ______________________ Date: ______________________

Instructions:

- Please circle the letter/number that you think is the best answer for a given question.
- Please do not answer a question if it makes you uncomfortable or you would not like to answer it for any other reason.

Personal Information

1) Gender
   a) Male
   b) Female

2) Age group
   a) 18-25
   b) 26-35
   c) 36-45
   d) 45-60

3) Race/Ethnicity
   a) Caucasian
   b) Black
   c) Native American
   d) Hispanic
   e) Asian/Pacific Islander
   f) Multicultural
   g) Other ______________________
   h) Do not wish to disclose

4) Nationality

   ____

5) Academic background
   a) Engineering
   b) Pure Sciences
   c) Social Sciences
   d) Architecture
   e) Business
   f) Education
   g) Other (Please Specify) ______________________

Computers

6) How long have you used computers?
   a) Less than 6 months

b) 6 months to a year
c) a year to two years
d) more than two years

7) How often do you use a computer?
a) Daily (almost everyday)
b) Weekly (2 to 3 times a week)
c) Monthly (2 to 3 times a month)
d) Less than once a month

8) What type of computer do you use? (please circle all that apply)
a) Apple Macintosh
b) Windows 95/Windows 98/Windows NT
c) Unix workstations (Sun, HP, Linux, SGI, etc.)
d) Other (Please Specify) ______________________________

9) What electronic items do you use? (please circle all that apply)
a) Desktop Computer
b) Notebook Computer
c) PDA (Personal Digital Assistant)
d) Cell phone (with color display and realistic musical sounds)
e) Cell phone (with color display)
f) Cell phone (with black-and-white display)
g) Blackberry e-mail device

Cameras

10) How long have you used film cameras?
a) Less than 6 months
b) 6 months to a year
c) a year to two years
d) more than two years

11) How often do you use a film camera?
a) Daily (almost everyday)
b) Weekly (2 to 3 times a week)
c) Monthly (2 to 3 times a month)
d) Less than once a month

12) What kind of photographs do you click? (circle all that apply)
a) People at family gatherings and parties
b) My family
c) Artifacts that interest me
d) Memories of my travels
e) Nature

13) What kind of a camera do (did) you use? (circle all that apply)
a) Automatic point-and-shoot
b) Manual/SLR
c) Other ______________________________

14) What do you do with photographs? (circle all that apply)

a) View them
b) Scan pictures for emailing them to friends
c) Develop my own pictures
d) Develop pictures for the whole wide world

15) How do you rate your understanding of photography?

Novice		Amateur		Intermed iate		Advance d		Expert/ Master
↓		↓		↓		↓		↓
1	2	3	4	5	6	7	8	9

Digital Cameras

16) How long have you used digital cameras?
a) Never
b) Less than 6 months
c) 6 months to a year
d) a year to two years
e) more than two years

17) How often do (or did) you use a digital camera?
a) Daily (almost everyday)
b) Weekly (2 to 3 times a week)
c) Monthly (2 to 3 times a month)
d) Less than once a month

18) What do you do with digital photographs? (circle all that apply)
a) Transfer pictures from digital camera to a computer
b) Transfer pictures from computer to a Web site for sharing with others
c) Scale and down-sample images for quicker download
d) Modify digital images to improve the picture
e) Modify or combine digital images to add special effects

19) How do you rate your understanding of digital photography?

Novice		Amateur		Intermediate		Advanced		Expert/ Master
↓		↓		↓		↓		↓
1	2	3	4	5	6	7	8	9

Digital Images

Have you created digital images?
a) No, thank you. I just download or view them

b) Yes, Schematics or line diagrams only
c) Yes, I combine various digital images to create collages
d) Yes, I modify digital pictures
e) Yes, I create digital art or animation
f) Yes, I create these and/or other kinds of digital images

20) How long have you created computer images?
a) Less than 6 months
b) 6 months to a year
c) a year to two years
d) more than two years

21) How often do you create computer images?
a) Daily (almost everyday)
b) Weekly (2 to 3 times a week)
c) Monthly (2 to 3 times a month)
d) Less than once a month

22) How do you rate your expertise in creating or editing digital images?

Novice		I can find my way around		I can hold my own		Advanced		Expert/ Master
↓		↓		↓		↓		↓
1	2	3	4	5	6	7	8	9

Information Element Relationship Evaluation

During-Task Questionnaire

Alias Used: ______________________________ Date: __________________________

Instructions:

- Please circle the option that best fits your answer for a given question.
- Please feel free to skip any questions that make you uncomfortable or you would not like to answer for any other reason.

1. This level of similarity between image1 and image2 was

Completely distinct				**Neither similar nor dissimilar**				**Virtually Identical**
↓				↓				↓
1	2	3	4	5	6	7	8	9

2. If image1 was not available and had to be replaced by image2, I would

   a) let a software program do it **automatically (do not answer question 3)**
   b) like the software program to **ask me before replacing the image (please also answer question 3)**

3. For this pair of images, if the software program asked for my permission to replace image1 with image2, I would

Definitely **NOT approve** the change		**Probably NOT approve** the change		I'm not sure either way		**Probably approve** the change		Definitely **approve** the change
↓		↓		↓		↓		↓
1	2	3	4	5	6	7	8	9

Information Element Relationship Evaluation

Post-Task Questionnaire

Alias Used: ______________________ Date: ______________________

Instructions:

- Please feel free to skip any questions that make you uncomfortable or you would not like to answer for any other reason.

1. Did you use the additional information about properties of the documents (provided below some of the image/text/audio/video files) while making your decisions about their similarity?

    a) Not at all
    b) Yes, in some cases
    c) Yes, in most cases
    d) Wherever it was available

2. If you used the additional information, how did this information affect your perception of the similarity of the two representations?

3. If you used the additional information, how did this information affect your decision about replacing a document with the other?

4. In the cases where the additional information was not displayed, did you attempt to get additional information about the documents? What strategies did you use?

VITA

Name: Unmil Purushottam Karadkar

Address: Vrundavan Housing Complex , Sector D
A Wing, Flat No. 1, Opposite Guru Ganesh Nagar
Kothrud, Pune 411038
Maharashtra State, India

Email: unmil@csdl.tamu.edu

Education: B.E., Computer Engineering, University of Pune, 1996
M.S., Computer Science, Texas A&M University, 2000
Ph.D., Computer Science, Texas A&M University, 2011